

**PREDIKSI NILAI TUKAR RUPIAH TERHADAP DOLAR AMERIKA
DENGAN MENGGUNAKAN ALGORITME GENETIKA –
*BACKPROPAGATION***

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Dwi Novi Setiawan
NIM: 135150200111007



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

PREDIKSI NILAI TUKAR RUPIAH TERHADAP DOLAR AMERIKA DENGAN
MENGUNAKAN ALGORITME GENETIKA - *BACKPROPAGATION*

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh:
Dwi Novi Setiawan
NIM: 135150200111007

Skripsi ini telah diuji dan dinyatakan lulus pada
13 April 2018
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Candra Dewi, S.Kom., M.Sc.
NIP. 19771114 200312 2 001

Dosen Pembimbing II



Sigit Adinugroho, S.Kom., M.Sc.
NIK. 201607 880701 1 000

Mengetahui

Ketua Jurusan Teknik Informatika




Tri Astoto Kurniawan, S.T., M.T., Ph.D.
NIP. 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 13 April 2018



Dwi Novi Setiawan

NIM: 135150200111007

KATA PENGANTAR

Puji Syukur kehadiran Allah SWT yang telah memberikan rahmat dan karunia-Nya, sehingga Penulis dapat menyelesaikan skripsi yang berjudul “Prediksi Nilai Tukar Rupiah Terhadap Dolar Amerika Dengan Menggunakan Metode Algoritme Genetika - *Backpropagation*” dengan baik. Skripsi ini disusun untuk memenuhi salah satu syarat dalam mendapatkan gelar Sarjana Komputer di Fakultas Ilmu Komputer Universitas Brawijaya Malang.

Penulisan skripsi ini tidak lepas dari bimbingan serta dukungan dari berbagai pihak, baik secara moril maupun materiil. Oleh sebab itu, Penulis mengucapkan banyak terima kasih kepada:

1. Bapak Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D. selaku Dekan Fakultas Ilmu Komputer, Universitas Brawijaya Malang.
2. Bapak Tri Astoto Kurniawan, S.T., M.T., Ph.D. selaku Ketua Jurusan Teknik Informatika, Universitas Brawijaya Malang.
3. Ibu Candra Dewi, S.Kom., M.Sc. selaku dosen pembimbing I yang telah memberikan waktu dan nasehatnya, serta dengan sangat sabar membimbing selama penulisan skripsi.
4. Bapak Sigit Adinugroho S.Kom., M.Sc. selaku dosen pembimbing II yang telah memberikan waktu dan nasehatnya, serta dengan sangat sabar membimbing selama penulisan skripsi.
5. Bapak dan Ibu dosen yang telah memberikan ilmu selama Penulis menempuh pendidikan di Fakultas Ilmu Komputer Universitas Brawijaya.
6. Segenap karyawan di Fakultas Ilmu Komputer Universitas Brawijaya yang membantu Penulis dalam pelaksanaan skripsi ini.
7. Orang tua Penulis, Ibu Sumini, serta kakak-kakak Penulis, Yuni Widyastuti, Tri Nur Wijayanti, Dwi Nova Setiadi dan keluarga besar yang selalu mendukung dalam doa, memberikan tauladan dan kasih sayang, serta menjadi penyemangat dikala lelah dan semua itu semata-mata hanya untuk keberhasilan Penulis.
8. Sahabat terdekat penulis Muhammad Ridha Anshari yang terus memberikan dukungan, dan bantuan sehingga Penulis dapat menyelesaikan skripsi ini.
9. Rekan-rekan kerja di Eat Two Burger Bar yang terus memberikan semangat, dukungan, bantuan, serta canda tawa.
10. Teman-teman seperjuangan Brilliant Aristyo R, M Hafidzar, Tony Faqih P, Reiza Adi, Iman Aidil N, Muhammad Wildan Haq, Muchibudin Abas, Ahmad Rizal M, Muhammad Arif W, Yeara Taufan R, dan lainnya yang tidak bisa disebutkan satu persatu. Terima kasih atas doa, dukungan, bantuan, kebersamaan, dan canda tawanya.

Penulis menyadari bahwa dalam penulisan skripsi ini masih terdapat banyak kekurangan. Oleh sebab itu, penulis mengharapkan kritik dan saran yang membangun, sehingga skripsi ini bisa menjadi lebih baik lagi.

Malang, 13 April 2018

Penulis

Dn.setiawan39@gmail.com



ABSTRAK

Nilai tukar (kurs) merupakan nilai mata uang suatu negara yang dinyatakan ke dalam bentuk mata uang negara lain. Nilai tukar berperan penting dalam perdagangan internasional. Untuk menjaga kestabilan nilai tukar rupiah, maka pemerintah perlu memberlakukan kebijakan yang tepat. Oleh karena itu diperlukan sebuah algoritme prediksi yang mampu mengenali pola dari perubahan nilai tukar rupiah. Metode *Backpropagation* merupakan salah satu metode yang mampu mengenali pola pada data *time series*, sedangkan Algoritme Genetika merupakan metode yang mampu melakukan eksplorasi solusi yang lebih luas bagi *Backpropagation*. Pada Algoritme Genetika, bobot *Backpropagation* direpresentasikan dalam bentuk *real-code*. Implementasi Algoritme Genetika – *Backpropagation* memiliki tahapan inisialisasi populasi, *crossover*, mutasi, pelatihan setiap individu menggunakan *Backpropagation*, evaluasi, dan seleksi. Dari pengujian yang telah dilakukan didapatkan parameter yang paling optimum bagi Algoritme Genetika – *Backpropagation* jumlah generasi 90, ukuran populasi sebesar 20, *crossover rate* 0,1, *mutation rate* 0,9, jumlah *neuron* pada lapisan tersembunyi 13, *learning rate* 1 dan jumlah iterasi pelatihan *Backpropagation* sebesar 500. Hasil pengujian yang telah dilakukan didapatkan nilai MAPE terbaik sebesar 1,575318 dan rata-rata MAPE sebesar 1,741747. Algoritme juga mampu melakukan validasi dengan MAPE terbaik sebesar 1.0004917 dan rata-rata MAPE sebesar 1,077603. Algoritme Genetika mampu meningkatkan akurasi *Backpropagation* hingga 5,947759%.

Kata kunci: Nilai Tukar, Kurs, Algoritme Genetika, Jaringan Syaraf Tiruan, *Backpropagation*, Algoritme Genetika - *Backpropagation*

ABSTRACT

The exchange rate is the value of the currency of a country which is expressed in the form of currency of another country. Exchange rate has an important role in international trade. To maintain the stability of the rupiah exchange rate, the government needs to enact the right policy. Therefore, a prediction algorithm that is able to recognize the pattern of exchange rate changes is needed. Backpropagation is one of method that is able to recognize patterns in time series data, while Genetic Algorithm is one of the capable method to exploring wider solutions for Backpropagation. In the Genetic Algorithm, the weight of Backpropagation is represented in real-code. Implementation of Genetic Algorithm - Backpropagation has initialization phase of population, crossover, mutation, individual training using Backpropagation, evaluation, and selection. The most optimum parameters for Genetic Algorithm - Backpropagation are in 90 generation, 20 population size, 0.1 crossover rate, 0.9 mutation rate, number of neurons in hidden layer 13, learning rate 1 and number of iteration of Backpropagation training were 500. The results of the tests that have been done got the best MAPE value of 1.575318 and the average MAPE of 1.741747. The algorithm is also capable of performing validation with the best MAPE of 1.0004917 and the average MAPE of 1.077603. Genetic Algorithm can increase the accuracy of Backpropagation up to 5.947759%.

Keyword: *Exchange Rate, Genetic Algorithm, Artificial Neural Network, Backpropagation, Genetic Algorithm – Backpropagation*

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xiv
DAFTAR LAMPIRAN	xvi
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan	3
1.4 Manfaat.....	3
1.5 Batasan Masalah.....	3
1.6 Sistematika Pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 Dasar Teori.....	9
2.3 Nilai Tukar (Kurs).....	9
2.3.1 Faktor-Faktor yang Mempengaruhi Nilai Tukar	10
2.4 Inflasi.....	10
2.5 Tingkat Suku Bunga.....	11
2.6 Normalisasi	11
2.6.1 Normalisasi Min-Max	11
2.7 Jaringan Syaraf Tiruan.....	11
2.7.1 Arsitektur Jaringan Syaraf Tiruan	12
2.7.2 Fungsi Aktivasi	12
2.8 Jaringan Syaraf Tiruan <i>Backpropagation</i>	12
2.8.1 Pelatihan <i>Backpropagation</i>	13

2.9 Algoritme Genetika	15
2.9.1 Penentuan Parameter	16
2.9.2 Representasi Kromosom	16
2.9.3 Inisialisasi Populasi	17
2.9.4 Reproduksi	17
2.9.5 Evaluasi	18
2.9.6 Seleksi	18
2.9.7 Kriteria Penghentian	18
2.10 Pengukuran Performa	19
2.10.1 MAPE	19
BAB 3 METODOLOGI	20
3.1 Studi Literatur	20
3.2 Pengumpulan Data	21
3.3 Analisis Kebutuhan	21
3.4 Perancangan	22
3.4.1 Deskripsi Sistem	22
3.4.2 Arsitektur Backpropagation	22
3.5 Pengujian Algoritme	23
3.6 Pengambilan Kesimpulan dan Saran	23
BAB 4 ALGORITME DAN PERANCANGAN	24
4.1 Formulasi Permasalahan	24
4.2 Normalisasi Min Max	26
4.3 Siklus Penentuan Bobot Optimum <i>Backpropagation</i> dengan Algoritme Genetika	30
4.3.1 Representasi Kromosom	31
4.3.2 Inisialisasi Populasi Awal	32
4.3.3 Reproduksi	34
4.3.4 Pelatihan Individu dengan <i>Backpropagation</i>	38
4.3.5 Evaluasi	43
4.3.6 Seleksi	47
4.4 Perancangan Antarmuka	51

4.4.1 Antarmuka Proses Optimasi Algoritme Genetika – <i>Backpropagation</i>	51
4.4.2 Antarmuka Menampilkan Data	52
4.4.3 Antarmuka Menampilkan Hasil Pengujian	52
4.5 Perancangan Pengujian	53
4.5.1 Pengujian Jumlah Generasi	54
4.5.2 Pengujian Ukuran Populasi	54
4.5.3 Pengujian Nilai <i>Crossover Rate</i> dan <i>Mutation Rate</i>	55
4.5.4 Pengujian Jumlah <i>Neuron</i> pada Lapisan Tersembunyi <i>Backpropagation</i>	56
BAB 5 PEMBAHASAN	57
5.1 Lingkungan Implementasi	57
5.1.1 Lingkungan Perangkat Keras	57
5.1.2 Lingkungan Perangkat Lunak	57
5.2 Deskripsi Data	57
5.3 Kode Program Implementasi Algoritme Genetika – <i>Backpropagation</i>	57
5.3.1 Fungsi Inisialisasi Populasi	58
5.3.2 Fungsi Reproduksi <i>Crossover</i>	58
5.3.3 Fungsi Reproduksi Mutasi	59
5.3.4 Fungsi Pelatihan Individu dengan <i>Backpropagation</i>	60
5.3.5 Fungsi Evaluasi	61
5.3.6 Fungsi Seleksi	62
5.4 Implementasi Antarmuka	64
5.4.1 Tab Data	64
5.4.2 Tab Pelatihan <i>Backpropagation</i>	64
5.4.3 Tab Pelatihan Algoritme Genetika – <i>Backpropagation</i>	65
5.4.4 Tab Pengujian dan Hasil	65
BAB 6 PENGUJIAN DAN ANALISIS	67
6.1.1 Pengujian Maksimum Iterasi	67
6.1.2 Pengujian Nilai <i>Learning Rate (Alpha)</i>	69
6.1.3 Pengujian Jumlah <i>Neuron</i> pada Lapisan Tersembunyi	70
6.2 Validasi dan Pengujian <i>Backpropagation</i>	72

6.3 Pelatihan Algoritme Genetika – <i>Backpropagation</i>	73
6.3.1 Pengujian Jumlah Generasi	73
6.3.2 Pengujian Ukuran Populasi	75
6.3.3 Pengujian <i>Crossover Rate</i> dan <i>Mutation Rate</i>	77
6.3.4 Pengujian Maksimum Iterasi.....	78
6.3.5 Pengujian <i>Learning Rate (Alpha)</i>	80
6.3.6 Pengujian Jumlah <i>Neuron</i> pada Lapisan Tersembunyi	81
6.4 Validasi dan Pengujian Algoritme Genetika – <i>Backpropagation</i>	83
6.5 Perbandingan Hasil Pengujian	84
BAB 7 PENUTUP	86
7.1 Kesimpulan.....	86
7.2 Saran	86
DAFTAR PUSTAKA.....	87
LAMPIRAN A DATA LATIH.....	89
LAMPIRAN B DATA UJI	93

DAFTAR TABEL

Tabel 4.1 Contoh data latih permasalahan	26
Tabel 4.2 Contoh data uji permasalahan	27
Tabel 4.3 Hasil normalisasi contoh data latih	29
Tabel 4.4 Hasil normalisasi contoh data uji	29
Tabel 4.5 Individu awal	33
Tabel 4.6 Hasil prediksi <i>Parent 1</i>	45
Tabel 4.7 Hasil denormalisasi prediksi <i>Parent 1</i>	46
Tabel 4.8 Hasil penghitungan MAPE dari <i>Parent 1</i>	46
Tabel 4.9 <i>Fitness</i> semua individu	47
Tabel 4.10 Nilai <i>fitness</i> kumulatif	49
Tabel 4.11 Nilai random	49
Tabel 4.12 Individu baru hasil seleksi	50
Tabel 4.13 Pengujian jumlah generasi	54
Tabel 4.14 Pengujian ukuran populasi	55
Tabel 4.15 Pengujian nilai <i>crossover rate</i> dan <i>mutation rate</i>	55
Tabel 4.16 Pengujian jumlah <i>neuron</i> pada lapisan tersembunyi <i>Backpropagation</i>	56
Tabel 6.1 Hasil pengujian maksimum iterasi pada pelatihan <i>Backpropagation</i> ...	67
Tabel 6.2 Hasil pengujian nilai <i>learning rate</i> pada pelatihan <i>Backpropagation</i> ...	69
Tabel 6.3 Hasil pengujian jumlah <i>neuron</i> pada pelatihan <i>Backpropagation</i>	71
Tabel 6.4 Hasil validasi dan pengujian akurasi pada pelatihan <i>Backpropagation</i>	73
Tabel 6.5 Hasil pengujian jumlah generasi pada pelatihan Algoritme Genetika - <i>Backpropagation</i>	74
Tabel 6.6 Hasil pengujian ukuran populasi pada pelatihan Algoritme Genetika - <i>Backpropagation</i>	75
Tabel 6.7 Hasil pengujian <i>crossover rate</i> dan <i>mutation rate</i> pada pelatihan Algoritme Genetika - <i>Backpropagation</i>	77
Tabel 6.8 hasil pengujian maksimum iterasi pada pelatihan Algoritme Genetika - <i>Backpropagation</i>	79
Tabel 6.9 Pengujian <i>learning rate</i> pada pelatihan Algoritme Genetika - <i>Backpropagation</i>	80

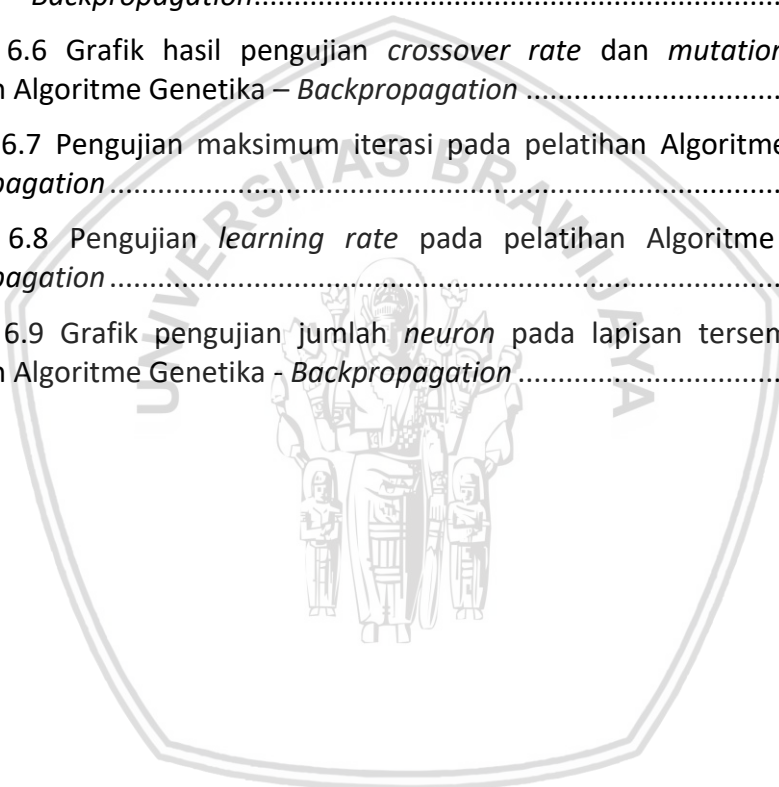
Tabel 6.10 Pengujian jumlah <i>neuron</i> pada lapisan tersembunyi pada pelatihan Algoritme Genetika - <i>Backpropagation</i>	82
Tabel 6.11 Hasil validasi dan pengujian akurasi pada pelatihan Algoritme Genetika – <i>Backpropagation</i>	84
Tabel 6.12 Perbandingan hasil validasi dan pengujian	85



DAFTAR GAMBAR

Gambar 3.1 Diagram alur metedologi penelitian	20
Gambar 3.2 Arsitektur <i>Backpropagation</i>	23
Gambar 4.1 Diagram alur sistem	24
Gambar 4.2 Diagram alur normalisasi	28
Gambar 4.3 Diagram alur Algoritme Genetika dan <i>Backpropagation</i>	30
Gambar 4.4 Representasi kromosom <i>real-code</i> dengan 2 <i>neuron</i> pada lapisan tersembunyi	31
Gambar 4.5 Diagram alur <i>crossover</i>	34
Gambar 4.6 Kromosom dari <i>Parent</i> 4	35
Gambar 4.7 Kromosom dari <i>Parent</i> 9	35
Gambar 4.8 Alpha untuk <i>crossover</i> antara <i>Parent</i> 4 dan <i>Parent</i> 9	35
Gambar 4.9 Kromosom dari <i>Child</i> 1 hasil <i>crossover</i>	36
Gambar 4.10 Kromosom dari <i>Child</i> 2 hasil <i>crossover</i>	36
Gambar 4.11 Diagram alur mutasi	37
Gambar 4.12 Kromosom dari <i>Parent</i> 8	37
Gambar 4.13 Kromosom dari <i>Child</i> 3 hasil mutasi.....	38
Gambar 4.14 Diagram alur pelatihan <i>Backpropagation</i>	39
Gambar 4.15 Diagram alur <i>Feedforward</i>	40
Gambar 4.16 Kromosom dari <i>Parent</i> 1	41
Gambar 4.17 Contoh data yang dihitung.....	41
Gambar 4.18 Diagram alur evaluasi	44
Gambar 4.19 Diagram alur denormalisasi	45
Gambar 4.20 Diagram alur seleksi <i>Roulette Wheel</i>	48
Gambar 4.21 Antar Proses Optimasi <i>Backpropagation</i> -Algoritma Genetika	51
Gambar 4.22 Antarmuka Menampilkan Data	52
Gambar 4.23 Antarmuka Hasil Pengujian	53
Gambar 5.1 Tab Data	64
Gambar 5.2 Tab Pelatihan <i>Backpropagation</i>	64
Gambar 5.3 Tab Pelatihan Algoritme Genetika - <i>Backpropagation</i>	65
Gambar 5.4 Tab Hasil	66

Gambar 6.1 Grafik hasil pengujian maksimum iterasi pada pelatihan <i>Backpropagation</i>	68
Gambar 6.2 Grafik hasil pengujian <i>learning rate</i> pada pelatihan <i>Backpropagation</i>	70
Gambar 6.3 Grafik hasil pengujian jumlah <i>neuron</i> pada pelatihan <i>Backpropagation</i>	72
Gambar 6.4 Grafik Hasil pengujian jumlah Generasi pada pelatihan Algorime Genetika – <i>Backpropagation</i>	75
Gambar 6.5 Grafik Hasil pengujian ukuran populasi pada pelatihan Algorime Genetika – <i>Backpropagation</i>	76
Gambar 6.6 Grafik hasil pengujian <i>crossover rate</i> dan <i>mutation rate</i> pada pelatihan Algoritme Genetika – <i>Backpropagation</i>	78
Gambar 6.7 Pengujian maksimum iterasi pada pelatihan Algoritme Genetika - <i>Backpropagation</i>	79
Gambar 6.8 Pengujian <i>learning rate</i> pada pelatihan Algoritme Genetika - <i>Backpropagation</i>	81
Gambar 6.9 Grafik pengujian jumlah <i>neuron</i> pada lapisan tersembunyi pada pelatihan Algoritme Genetika - <i>Backpropagation</i>	83



DAFTAR LAMPIRAN

LAMPIRAN A DATA LATIH.....	89
LAMPIRAN B DATA UJI	93



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Uang dapat didefinisikan sebagai sesuatu yang secara umum dapat diterima sebagai alat pembayaran yang sah. Pada dasarnya uang merupakan alat yang mampu digunakan sebagai alat pembayaran barang, jasa, ataupun kekayaan berharga dan juga sebagai alat pembayaran hutang. Umumnya setiap negara memiliki mata uang yang berbeda-beda, mata uang yang digunakan di suatu negara akan berbeda dengan mata uang di negara lainnya. Mata uang ini memiliki fungsi sebagai alat pembayaran yang sah dalam melakukan aktivitas ekonomi di negara tersebut. Sedangkan untuk melakukan aktivitas ekonomi pada negara lain perlu melakukan pertukaran mata uang kedalam bentuk mata uang negara tersebut, hal inilah yang disebut pasar valuta asing. Nilai mata uang suatu negara terhadap negara lain dalam pasar valuta asing selalu berubah-ubah dan tidak tetap. Hal ini menyebabkan perubahan dalam pasar valuta terlihat seakan terjadi secara acak (Shioda, Deng & Sakurai, 2011). Termasuk di dalamnya mata uang yang nilai tukarnya tidak menentu dalam pasar valuta asing.

Nilai tukar rupiah terhadap dolar Amerika menjadi perhatian tersendiri, dimana nilai tukar rupiah akan mempengaruhi harga barang pokok yang diimpor maupun nilai jual barang yang diekspor ke Amerika yang ahirnya akan mempengaruhi kesejahteraan masyarakat. Data dari Bank Indonesia menunjukkan terjadinya fluktuasi nilai tukar rupiah terhadap dolar Amerika yang begitu beragam namun cenderung menurun di setiap harinya (Subiantara, 2015). Bahkan anjloknya nilai tukar rupiah terhadap dolar Amerika sempat terjadi di bulan September tahun 2015 yang menembus angka Rp 14.470 dan tercatat sebagai yang terburuk sejak tahun 1988. Anjloknya nilai tukar rupiah berimbas pada kebutuhan pokok masyarakat. Muchlas (2015) dalam penelitiannya menyebutkan bahwa ekspor dalam negeri dipengaruhi oleh inflasi yang terjadi, jika inflasi meningkat maka dapat menyebabkan menurunnya kemampuan ekspor yang berakibat pada penurunan *supply* valuta asing di dalam negeri. Penurunan Jumlah Uang yang Beredar juga akan mempengaruhi nilai valuta asing yang semakin menurun. Selain itu tingkat suku bunga yang tinggi mampu menarik investor asing untuk melakukan investasi di dalam negeri. Dengan melihat keadaan tersebut, penelitian terhadap prediksi nilai tukar rupiah dirasa perlu dilakukan.

Terdapat banyak faktor yang mempengaruhi terjadinya perubahan nilai tukar rupiah terhadap dolar Amerika. Muchlas (2015) pada penelitiannya membuktikan bahwa inflasi, Jumlah Uang Beredar dan suku bunga memiliki hubungan dengan nilai tukar rupiah terhadap dolar Amerika. Selain penelitian tersebut, Wibowo & Amir (2006) juga telah membuktikan bahwa terdapat pengaruh antara nilai tukar rupiah terhadap dolar Amerika, tingkat suku bunga, dan jumlah uang yang beredar pada bulan sebelumnya akan mempengaruhi nilai tukar rupiah terhadap dolar Amerika pada bulan selanjutnya.

Menurut Suhendra dan Wardoyo (2015) Jaringan Syaraf Tiruan mampu menyelesaikan masalah klasifikasi dengan skala masalah yang cukup luas namun terkendala dengan kemungkinan terjadinya konvergensi yang buruk dan tidak stabil karena terjebak dalam lokal optimum. Algoritma Genetika merupakan salah satu metode yang dapat digunakan untuk menyelesaikan masalah pada Jaringan Syaraf Tiruan (Khan, 2008).

Penelitian berkaitan dengan prediksi nilai tukar rupiah terhadap dolar Amerika pernah dilakukan sebelumnya oleh Gernowo dan Ernitawati (2013). Penelitian yang dilakukan dengan menggunakan data dari Bank Indonesia selama 10 tahun terakhir yaitu dari tahun 2002 hingga 2013. Data yang digunakan merupakan data rata-rata setiap bulannya. Penelitian ini mengimplementasikan metode ANFIS yang merupakan salah satu metode dalam Jaringan Syaraf Tiruan yang dikembangkan menggunakan Fuzzy Sugeno. Metode ini mampu memberikan kebenaran hingga mencapai 89,51%. Dari penelitian ini dapat disimpulkan bahwa prediksi nilai tukar rupiah terhadap dolar Amerika dapat dilakukan dengan menggunakan data dan metode yang tepat (Gernowo & Ernitawati, 2013). Sumarto (2016) juga melakukan penelitian yang sama terhadap nilai tukar rupiah dan dolar Amerika dengan menggunakan metode *Support Vector Regression* yang dioptimasi menggunakan *Simulated Annealing-Genetic Algorithm*. Penelitian ini melakukan prediksi nilai tukar harian dan menghasilkan nilai MAPE yang cukup baik yaitu 1,115397 meskipun tanpa menggunakan fitur berupa data inflasi, dan jumlah uang beredar. Penelitian lainnya dilakukan oleh Pramesti (2016) yang menggunakan metode Fuzzy Mamdani dan dioptimasi menggunakan Algoritme Genetika. Penelitian ini memanfaatkan Algoritme Genetika untuk mencari batas himpunan keanggotaan yang paling mendekati optimal. Dengan menggunakan fitur inflasi, jumlah uang beredar, dan selisih kurs selama 4 bulan terakhir penelitian ini mampu menghasilkan nilai MAPE sebesar 1,20164.

Fais (2013) melakukan prediksi *Backpropagation* dengan menggunakan Algoritme Genetika pada *Backpropagation Neural Network* pada peramalan saham dengan tingkat akurasi rata-rata hingga 87%. Sehingga dapat disimpulkan optimasi *Backpropagation* dengan menggunakan Algoritme Genetika dirasa mampu memberikan hasil yang lebih baik pada hasil *Backpropagation*. Dengan pertimbangan seperti yang telah dipaparkan diatas, penelitian berkaitan dengan prediksi nilai tukar rupiah terhadap dolar Amerika dengan menggunakan metode *Backpropagation* yang pencarian bobotnya akan dioptimasi menggunakan Algoritme Genetika diharapkan mampu melakukan eksplorasi solusi yang lebih luas.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, masalah dalam penelitian ini dapat dirumuskan menjadi beberapa poin seperti berikut:

1. Bagaimana menerapkan metode Algoritme Genetika untuk melakukan optimasi pada metode *Backpropagation* pada kasus peramalan nilai tukar Rupiah terhadap dolar Amerika?
2. Bagaimana tingkat *error* yang didapatkan dalam peramalan nilai tukar Rupiah terhadap dolar Amerika dengan metode *Backpropagation* yang dioptimasi dengan menggunakan Algoritme Genetika?

1.3 Tujuan

Tujuan dalam penelitian ini dapat diuraikan seperti berikut:

1. Menerapkan metode *Backpropagation* dengan *Algoritme Genetika* pada masalah peramalan nilai tukar Rupiah Indonesia terhadap dolar Amerika.
2. Mengetahui tingkat *error* yang dihasilkan dari sistem prediksi nilai tukar Rupiah Indonesia terhadap dolar Amerika.

1.4 Manfaat

Penelitian ini diharapkan mampu memberikan manfaat bagaimana agar mendapatkan sebuah algoritme prediksi dengan kinerja yang optimal dari segi akurasi dengan menggunakan metode optimasi Algoritme Genetika pada metode *Backpropagation* pada nilai tukar Rupiah terhadap dolar Amerika. Diharapkan hasil dari penelitian ini bisa menjadi bahan pertimbangan dalam menentukan algoritme yang terbaik yang dapat diimplementasikan dalam melakukan prediksi nilai tukar mata uang.

1.5 Batasan Masalah

Agar permasalahan dalam penelitian yang telah dirumuskan dapat terfokus, maka penelitian ini dibatasi dalam beberapa hal:

1. Melakukan prediksi pada nilai tukar Rupiah terhadap dolar Amerika.
2. Data yang digunakan merupakan data inflasi, *BI rate* dan nilai tukar Rupiah terhadap dolar Amerika setiap bulannya mulai dari Januari 2006 hingga Juli 2016 yang terdapat di halaman resmi Bank Indonesia.

1.6 Sistematika Pembahasan

BAB I : Pendahuluan

Menjelaskan latar belakang penelitian, rumusan masalah di dalam penelitian, tujuan dan manfaat penelitian, serta batasan

masalah dan sistematika penulisan yang digunakan di dalam penelitian.

BAB II : Landasan Kepustakaan

Menjelaskan dasar teori dan kajian pustaka yang mendasari penelitian prediksi nilai tukar rupiah terhadap dolar Amerika dengan menggunakan metode Algoritme Genetika – *Backpropagation*

BAB III : Metodologi

Bab ini akan menjelaskan mengenai metode dan tahapan yang digunakan dalam menyelesaikan permasalahan dalam penelitian.

BAB IV : Algoritme dan Perancangan

Bab ini menjelaskan arsitektur pada *Backpropagation*, perancangan algoritma, penghitungan manual Algoritme Genetika – *Backpropagation*, perancangan antarmuka, dan perancangan pengujian.

BAB V : Pembahasan

Bab ini menjelaskan lingkungan implementasi yang meliputi lingkungan perangkat keras dan lingkungan perangkat lunak, kode program, implementasi Algoritme Genetika – *Backpropagation*, dan implementasi antarmuka.

BAB VI : Pengujian dan Analisis

Bab ini menjelaskan pelatihan Algoritme Genetika – *Backpropagation*, pengujian parameter *Backpropagation*, pengujian parameter Algoritme Genetika, dan perbandingan pengujian hasil pelatihan *Backpropagation* dengan Algoritme Genetika – *Backpropagation*.

BAB VII : Penutup

Bab ini menjelaskan kesimpulan dari hasil penelitian yang telah dilakukan dan saran untuk pengembangan lebih lanjut.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Pada bab 2 dijelaskan penelitin-penelitian terdahulu yang sebelumnya mengangkat masalah prediksi nilai tukar rupiah terhadap dolar Amerika maupun penggunaan *Backpropagation* yang dioptimasi dengan menggunakan Algoritme Genetika. Penelitian yang dijadikan kajian pustaka antara lain “Peramalan Nilai Tukar Rupiah (IDR) Terhadap dolar Amerika (USD) Menggunakan Metode *Support Vector Regression* (SVR) Dengan *Simulated Annealing – Genetic Algorithm* (SA-GA)” (Sumarto 2016), “Optimasi Fuzzy Inference System Mamdani Untuk Prediksi Nilai Tukar Rupiah Terhadap dolar Amerika Menggunakan Algoritme Genetika” (Pramesti, 2016), “Peramalan *Time Series* Saham Menggunakan *Backpropagation Neural Network* Berbasis Algoritma Genetika” (Huda, 2013).

Pada penelitian Sumarto (2016) telah dijelaskan bahwa nilai tukar rupiah merupakan masalah regresi yang nilainya selalu berubah-ubah dan tidak tetap. penulis melakukan penelitian berupa sistem peramalan nilai tukar rupiah (IDR) terhadap dolar Amerika yang menggunakan metode *Support Vector Regression* (SVR) dengan *Simulated Annealing – Genetic Algorithm* (SA-GA) dimana hasil yang diberikan terbilang cukup baik, yaitu dengan nilai MAPE 1,115397 dengan melakukan penelitian terhadap data dari halaman International Monetary Fund dari tahun 2013 hingga 2015 dengan menggunakan menggunakan nilai tukar rupiah 3 hari sebelumnya sebagai fitur masukan dan prediksi nilai tukar rupiah hari ini sebagai keluaran. Penelitian tersebut menggunakan *Support Vector Regression* sebagai metode yang mampu menyelesaikan masalah regresi pada prediksi nilai tukar mata uang tersebut. Selain itu penelitian yang sama juga dilakukan oleh Pramesti, (2016) dengan melakukan prediksi nilai tukar rupiah terhadap dolar Amerika dengan menggunakan metode *Fuzzy Inference* Mamdani yang dioptimasi dengan Algoritme Genetika. Penelitian tersebut memanfaatkan Algoritme Genetika untuk menentukan batas pada himpunan keanggotaan Fuzzy Inference Mamdani yang akan digunakan untuk melakukan prediksi nilai tukar rupiah terhadap dolar Amerika. Dengan menggunakan data dari Bank Indonesia, penelitian ini mampu memberikan hasil MAPE sebesar 1.20164.

Penelitian Al Huda (2013) menunjukkan bahwa Algoritme Genetika dan *Backpropagation* merupakan kombinasi metode yang cukup baik digunakan dalam melakukan peramalan data *time series* berupa saham. Pada penelitian ini Input yang digunakan berupa *open*, *high*, *low*, dan *close* data data *time series* dilatih menggunakan Algoritme Genetika yang kemudian hasil bobot awal terbaik akan latih akan dilatih lagi menggunakan *Backpropagation* untuk memastikan mendapatkan calon solusi terbaik yang diharapkan mampu memberikan hasil *global optimum*. Penelitian ini memiliki tingkat akurasi rata-rata 87%.

Judul	Objek	Metode	Keluaran
	Masukkan Parameter	Proses	Hasil Penelitian
Peramalan Nilai Tukar Rupiah (IDR) Terhadap dolar Amerika (USD) Menggunakan Metode <i>Support Vector Regression (SVR)</i> Dengan <i>Simulated Annealing – Genetic Algorithm (SA-GA)</i>	Data tukar rupiah terhadap dolar Amerika pada tanggal 1 Januari 2013 hingga 31 Desember 2015 dari dataset <i>International Monetary Fund</i>	SVR dengan optimasi <i>Simulated Annealing- Genetic Algorithm (SA-GA)</i>	<ul style="list-style-type: none"> - Parameter optimal - Nilai tukar rupiah yang diramalkan Nilai MAPE
	<ul style="list-style-type: none"> - Jumlah generasi - Ukuran populasi - Tingkat <i>crossover</i> - Tingkat mutasi - Temperatur awal - Temperatur akhir - <i>Cooling factor</i> - Jumlah iterasi SVR - Data latih - Data uji 	<p>Bagian SA:</p> <ul style="list-style-type: none"> - Inisialisasi populasi - Cek temperatur - Gunakan bagian GA - Evaluasi <i>fitness</i> - Kurangi temperature - Ulangi hingga temperature kurang dari atau sama dengan temperature akhir <p>Bagian GA:</p> <ul style="list-style-type: none"> - Ambil populasi dari SA - Lakukan operasi reproduksi, <i>crossover</i>, mutasi - Lakukan seleksi <p>Kembalikan populasi ke bagian SA</p>	Implementasi optimasi SA-GA terhadap SVR dapat menghasilkan nilai <i>error</i> yang kecil dalam peramalan nilai tukar rupiah (IDR) terhadap dolar Amerika dengan nilai MAPE 1.115397

Judul	Objek	Metode	Keluaran
	Masukkan Parameter	Proses	Hasil Penelitian
Optimasi <i>Fuzzy Inference System</i> Mamdani Untuk Prediksi Nilai Tukar Rupiah Terhadap dolar Amerika Menggunakan Algoritme Genetika	Data tukar rupiah terhadap dolar Amerika pada bulan Januari 2009 sampai Desember 2015 dari dataset Bank Indonesia	<i>Fuzzy Inference</i> Mamdani yang dioptimasi dengan Algoritme Genetika	<ul style="list-style-type: none"> - Individu hasil optimasi - Data aktual, data prediksi dan <i>error</i> dari seluruh data uji pada masing-masing individu pada generasi terakhir. - Total <i>fitness</i> dari individu terbaik. - Nilai <i>fitness</i> dan kromosom individu terbaik pada setiap generasinya .
	<ul style="list-style-type: none"> - Jumlah generasi - Ukuran populasi - Tingkat <i>crossover</i> - Tingkat mutasi 	Bagian Algoritme Genetika: <ul style="list-style-type: none"> - Inisialisasi populasi - Lakukan operasi reproduksi, <i>crossover</i>, dan mutasi - Lakukan seleksi - Gunakan individu terbaik hasil Algoritme Genetika kepada <i>Fuzzy Inference</i> Mamdani - Algoritme pada <i>Fuzzy Inference</i> Mamdani: - Gunakan hasil Algoritme Genetika sebagai batas pada setiap himpunan keanggotaan 	Implementasi optimasi <i>Fuzzy Inference</i> Mamdani mampu menghasilkan tingkat <i>error</i> yang kecil dalam melakukan peramalan nilai tukar rupiah terhadap dolar Amerika dengan nilai MAPE 1.20164

Judul	Objek	Metode	Keluaran
	Masukkan Parameter	Proses	Hasil Penelitian
Peramalan <i>Time Series</i> Saham Menggunakan <i>Backpropagation Neural Network</i> Berbasis Algoritma Genetika	Data <i>Time Series</i> Saham	Metode <i>Backpropagation</i> yang dioptimasi dengan menggunakan Algoritma Genetika	<i>Trade Volume</i>
	<i>Open, high, low, close.</i>	<ul style="list-style-type: none"> - Inisialisasi bobot awal - Pelatihan dengan <i>Backpropagation</i> - Pelatihan dengan Algoritma Genetika 	Implementasi <i>Backpropagation</i> dan Algoritma Genetika pada peramalan <i>time series</i> saham dengan parameter pelatihan terbaik <i>learning rate</i> 0.004, momentum 0.7, peluang <i>crossover</i> 0.7, peluang mutasi 0.1, jumlah populasi 100, dan jumlah generasi 1000 yang menghasilkan tingkat akurasi rata-rata sebesar 87%.

Judul	Objek	Metode	Keluaran
	Masukkan Parameter	Proses	Hasil Penelitian
Peramalan Nilai Tukar Rupiah Terhadap dolar Amerika Dengan Menggunakan Metode Algoritme Genetika - <i>Backpropagation</i>	Data nilai tukar rupiah terhadap dolar Amerika dari data set Bank Indonesia dari Januari 2006 hingga Juli 2016	<i>Backpropagation</i> yang dioptimasi dengan menggunakan Algoritme Genetika	<ul style="list-style-type: none"> - Nilai bobot optimum - Nilai tukar rupiah hasil prediksi - Nilai MAPE
	<ul style="list-style-type: none"> - Jumlah generasi pada Algoritme Genetika - Ukuran populasi pada Algoritme Genetika - Tingkat <i>crossover</i> pada Algoritme Genetika - Tingkat mutasi pada Algoritme Genetika - Jumlah lapisan tersembunyi pada arsitektur <i>Backpropagation</i> - Tingkat pelatihan pada <i>Backpropagation</i> - Jumlah iterasi pelatihan pada <i>Backpropagation</i> 	Bagian Algoritme Genetika: <ul style="list-style-type: none"> - Inisialisasi populasi - Lakukan operasi <i>crossover</i>, dan mutasi - Latih setiap individu dengan menggunakan <i>Backpropagation</i> - Evaluasi nilai <i>fitness</i> setiap Individu - Lakukan seleksi - Individu terbaik digunakan sebagai bobot pada prediksi nilai tukar rupiah terhadap dolar Amerika. 	Implementasi optimasi Algoritme Genetika terhadap <i>Backpropagation</i> dapat memberikan bobot optimum dan menghasilkan tingkat <i>error</i> yang kecil dalam peramalan nilai tukar rupiah terhadap dolar Amerika.

2.2 Dasar Teori

Teori dasar mengenai objek yang diteliti dan metode yang digunakan akan dijelaskan secara singkat pada sub bab 2.2. Objek dan metode yang menjadi bahan penelitian antara lain: Nilai Tukar, Jaringan Syaraf Tiruan, *Backpropagation*, dan Algoritme Genetika.

2.3 Nilai Tukar (Kurs)

Nilai tukar (kurs) menurut Kamus Besar Bahasa Indonesia (KBBI) merupakan nilai mata uang suatu negara yang dinyatakan ke dalam bentuk mata uang negara lain. Nilai tukar berperan penting dalam perdagangan antar negara. Fluktuasi nilai tukar dapat membahayakan operasi bisnis internasional (Bahramy & Crone, 2013). Menurut Adiningsih (dalam Mansur, 2009), penurunan kurs rupiah terhadap mata uang asing akan memberikan pengaruh negative pada

kondisi ekonomi Indonesia termasuk pasar modal. Peningkatan tingkat suku bunga akan mempengaruhi minat pemodal untuk melakukan investasi.

2.3.1 Faktor-Faktor yang Mempengaruhi Nilai Tukar

Dampak yang ditimbulkan oleh nilai tukar terhadap tingkat perekonomian suatu negara sangat besar. Peningkatan dan penurunan ekspor dan impor suatu negara juga dipengaruhi oleh nilai tukar, dimana semakin rendah nilai tukar mata uang maka menyebabkan semakin meningkatnya impor dan menurunnya ekspor, dan begitu pula sebaliknya. Terdapat beberapa faktor yang mempengaruhi perubahan pada nilai tukar mata uang menurut (Bergen, 2010), yaitu:

1. Perbedaan tingkat inflasi dua negara
2. Perbedaan tingkat suku bunga antara dua negara
3. Neraca perdagangan
4. Hutang publik
5. Rasio harga ekspor dan impor
6. Kestabilan politik dan ekonomi

2.4 Inflasi

Inflasi adalah suatu kondisi dimana harga barang akan terus mengalami kenaikan. Berbeda dengan inflasi, deflasi merupakan penurunan nilai uang secara terbuka yang diakibatkan oleh jumlah barang yang beredar terlalu banyak sehingga harga dari barang-barang tersebut menurun. Inflasi dapat dipengaruhi dari dalam maupun luar negeri. Contoh inflasi yang terjadi di dalam negeri adalah defisit anggaran belanja dimana uang terus dicetak namun kegagalan pasar yang mengakibatkan naiknya harga bahan makanan. Sementara itu, meningkatnya harga barang impor akan menyebabkan terjadinya inflasi dari luar negeri. Hal ini dapat dipengaruhi oleh tarif impor barang yang meningkat, ataupun memang terjadi kenaikan pada harga barang impor. Inflasi yang berakibat pada beberapa barang yang memiliki keterkaitan dapat disebut sebagai inflasi tertutup, sedangkan inflasi yang mampu mempengaruhi harga barang secara luas maka hal ini disebut sebagai inflasi terbuka (Fitriah & Abadi, 2011).

Kestabilan inflasi memiliki peran penting dalam pertumbuhan ekonomi yang berkesinambungan yang akhirnya akan memberikan peningkatan kesejahteraan masyarakat (Pramesti, 2016). Inflasi yang tinggi akan menyebabkan pendapatan yang menurun, standar hidup yang juga ikut menurun, dan akhirnya semua orang, terutama orang miskin bertambah miskin.

2.5 Tingkat Suku Bunga

Tingkat suku bunga merupakan persentase dari harga yang harus dibayarkan dari penggunaan uang pada jangka waktu tertentu. Tingkat suku bunga juga dapat diartikan sebagai harga yang harus dibayarkan atas pertukaran mata uang nantinya (Boediono. 1998).

BI rate merupakan suku bunga kebijakan yang menggambarkan kebijakan moneter yang ditetapkan oleh Bank Indonesia yang kemudian diumumkan ke masyarakat luas. *BI rate* merupakan hasil dari rapat Dewan Gubernur bulanan yang diumumkan oleh Dewan Gubernur Bank Indonesia melalui pengelolaan likuiditas di pasar uang agar sasaran operasional kebijakan moneter dapat tercapai. Dengan mempertimbangkan faktor-faktor dalam perekonomian, Bank Indonesia akan menaikkan *BI rate* apabila inflasi yang terjadi diperkirakan melebihi perkiraan, sebaliknya Bank Indonesia akan menurunkan *BI rate* bila dirasa inflasi ke depan kurang dari sasaran yang telah ditetapkan.

2.6 Normalisasi

Normalisasi merupakan salah satu proses transformasi data pada *preprocessing* dalam *Data Mining*. Normalisasi dilakukan dengan mengubah data yang kompleks ke dalam bentuk yang lebih sederhana namun tidak menghilangkan ataupun merubah isi dari data tersebut. Terdapat beberapa macam normalisasi data yang dapat dilakukan, diantaranya adalah normalisasi min max

2.6.1 Normalisasi Min-Max

Normalisasi ini dilakukan dengan mencari nilai minimum dan maksimum yang terdapat pada data, dan merubah data kedalam skala baru dengan minimum dan maksimum baru yang telah ditentukan. Normalisasi min-max dilakukan sesuai dengan Persamaan

$$data\ baru = \frac{(data - min) * (max\ baru - min\ baru)}{(max - min)} + min\ baru \quad (2.1)$$

Data baru = data yang telah dinormalisasi

Min = nilai minimum dari seluruh data

Max = nilai maksimum dari seluruh data

Min baru = nilai minimum baru yang telah ditentukan

Max baru = nilai maksimum baru yang telah ditentukan

2.7 Jaringan Syaraf Tiruan

Jaringan Syaraf Tiruan merupakan cabang dari Kecerdasan Buatan dengan metode atau algoritme yang bertujuan untuk mencerdaskan sistem dengan mampu melakukan pembelajaran terhadap suatu masalah. Jaringan Syaraf Tiraun merepresentasikan sebuah jaringan syaraf otak manusia ke dalam bentuk

sebuah fungsi matematis sehingga mampu beradaptasi dalam menyelesaikan masalah.

2.7.1 Arsitektur Jaringan Syaraf Tiruan

Terdapat beberapa arsitektur di dalam jaringan Syaraf Tiruan, diantaranya adalah sebagai berikut:

1. Jaringan dengan lapisan tunggal (*single layer net*)

Jaringan dengan lapisan tunggal hanya memiliki satu lapisan pada arsitekturnya dengan bobot yang langsung dihubungkan dengan masukan yang kemudian diolah menjadi keluaran.

2. Jaringan dengan banyak lapisan (*multilayer net*)

Jaringan dengan banyak lapisan memiliki lapisan diantara masukan dan keluaran, dengan bobot yang berbeda sesuai dengan jumlah lapisan tersembunyi dari lapisan tersebut.

2.7.2 Fungsi Aktivasi

Fungsi aktivasi merupakan bentuk dari operasi matematika yang diberikan pada sinyal output pada jaringan syaraf tiruan. Fungsi aktivasi memiliki fungsi untuk membangkitkan nilai hasil dari sebuah *neuron* pada jaringan syaraf tiruan.

2.7.2.1 Fungsi Aktivasi *Sigmoid Biner*

Fungsi aktivasi *sigmoid biner* merupakan fungsi aktivasi yang paling umum digunakan pada jaringan syaraf tiruan dengan metode *Backpropagation*. Jaringan syaraf tiruan yang memiliki interval output antara 0 hingga 1 sering menggunakan fungsi aktivasi ini karena fungsi aktivasi *sigmoid biner* memberikan hasil antara 0 hingga 1. Secara matematis, fungsi aktivasi *sigmoid biner* dituliskan seperti Persamaan 2.2.

$$y = f(x) = \frac{1}{1 + e^{-\sigma x}} \quad (2.2)$$

2.8 Jaringan Syaraf Tiruan *Backpropagation*

Backpropagation merupakan salah satu metode dalam Jaringan Syaraf Tiruan yang cukup akurat dalam melakukan peramalan terhadap data *time series*. Metode *Backpropagation* merupakan metode dalam Jaringan Syaraf Tiruan yang terdiri dari multi lapisan dan menggunakan propagasi mundur sebagai alur pembelajaran. Menurut Pratama, (1999) *Backpropagation* merupakan algoritme yang menggunakan pendekatan *steepest descent*. *Backpropagation* memiliki perbedaan mendasar dengan Jaringan Syaraf Tiruan yang lain, yaitu pada proses pembelajarannya. Pada *Backpropagation* pembelajaran dilakukan dengan melihat nilai *error* pada sistem yang kemudian menjadi variabel yang digunakan sebagai penyesuaian bobot yang digunakan. Metode ini banyak digunakan sebagai metode peramalan, prediksi dan pengenalan dengan akurasi yang cukup baik (Dewi, C. Muslikh, M. 2013).

Backpropagation memiliki perbedaan mendasar dengan Jaringan Syaraf Tiruan yang lain, yaitu pada proses pembelajarannya. Pada *Backpropagation* pembelajaran dilakukan dengan melihat nilai *error* pada sistem yang kemudian menjadi variabel yang digunakan sebagai penyesuaian bobot yang digunakan.

2.8.1 Pelatihan *Backpropagation*

Pelatihan *Backpropagation* menurut Siang, J. J. (2005) terdiri dari tiga fase, yaitu:

1. Fase I: *Feedforward*

Merupakan fase dimana data masukan dilakukan proses matematika di setiap lapisan sehingga menghasilkan keluaran. Keluaran akan menjadi acuan dalam penentuan perbaikan yang dilakukan pada langkah selanjutnya.

2. Fase II: *Backpropagation*

Menghitung nilai sensitifitas pada setiap lapisan. Dimana sensitifitas setiap lapisan m dihitung dari sensitifitas pada lapisan $m+1$, sehingga perhitungan sensitifitas berjalan mundur.

3. Fase III: *Weight Update*

Perbaikan terhadap nilai bobot (w) dan (v) dengan menggunakan pendekatan *steepest descent*.

Berikut langkah-langkah yang dilakukan dalam proses pembelajaran *Backpropagation*.

Langkah 1

Inisialisasi bobot dengan menggunakan bilangan acak

Langkah 2

Ulangi langkah pelatihan hingga kondisi iterasi terpenuhi

Langkah 3

Untuk setiap pasang data latih, lakukan langkah 4 hingga langkah 9

2.8.1.2 Fase I: *Feedforward*

Langkah 4

Setiap node masukan ($x_i, i = 1, \dots, n$) menerima sinyal masukan dan meneruskan masukan x_i ke node-node lapisan tersembunyi.

Langkah 5

Hitung semua keluaran pada node lapisan tersembunyi z_j , ($j = 1, \dots, p$)

$$z_{net_j} = v_{j0} + \sum_{i=1}^n x_i \cdot v_{ji} \quad (2.3)$$

$$z_j = f(x_{net_j}) = \frac{1}{1 + \exp(-z_{net_j})} \quad (2.4)$$

Langkah 6

Hitung semua keluaran pada node y_k ($k = 1, \dots, m$)

$$y_{net_k} = w_{k0} + \sum_{j=1}^p z_j \cdot w_{kj} \quad (2.5)$$

$$y_k = f(y_{net_k}) = \frac{1}{1 + \exp(-y_{net_k})} \quad (2.6)$$

2.8.1.3 Fase II: *Backpropagation*

Langkah 7

Hitung faktor δ pada node keluaran berdasarkan kesalahan hasil pada setiap node keluaran y_k dimana $k = 1, \dots, m$

$$\delta_k = (t_k - y_k) f'(y_{net_k}) = (t_k - y_k) y_k (1 - y_k) \quad (2.7)$$

δ_k merupakan nilai kesalahan yang digunakan dalam perubahan bobot

Hitung perubahan bobot w_{kj} dengan tingkat pembelajaran α

$$\Delta w_{kj} = \alpha \delta_k z_j \quad (2.8)$$

Dimana $k = 1, \dots, m$ dan $j = 0, \dots, p$

Langkah 8

Hitung faktor δ lapisan tersembunyi berdasarkan kesalahan pada setiap lapisan tersembunyi z_j dimana $j = 1, \dots, p$

$$\delta_{net_j} = \sum_{k=1}^m \delta_k \cdot w_{kj} \quad (2.9)$$

Faktor δ lapisan tersembunyi:

$$\delta_j = \delta_{net_j} \cdot f'(z_{net_j}) = \delta_{net_j} \cdot z_j (1 - z_j) \quad (2.10)$$

Hitung perubahan bobot v_{ji}

$$\Delta v_{ji} = \alpha \delta_j x_i \quad (2.11)$$

Dimana $j = 1, \dots, p$ dan $i = 0, \dots, n$

2.8.1.4 Fase III: *Weight Update*

Langkah 9

Lakukan penghitungan pada perubahan untuk bobot pada node keluaran

$$w_{kj}(\text{baru}) = w_{kj}(\text{lama}) + \Delta w_{kj} \quad (2.12)$$

Dimana $k = 1, \dots, p$ dan $i = 0, \dots, p$

Hitung perubahan bobot garis yang menuju lapisan tersembunyi

$$v_{ji}(\text{baru}) = v_{ji}(\text{lama}) + \Delta v_{ji} \quad (2.13)$$

Dimana $j = 1, \dots, p$ dan $i = 0, \dots, n$

2.9 Algoritme Genetika

Algoritme Genetika merupakan metode dalam Algoritme Evolusi. Dari beberapa teknik optimasi dalam algoritme evolusi, Algoritme Genetika merupakan teknik optimasi yang paling populer dan terbukti mampu memecahkan permasalahan-permasalahan yang kompleks seperti yang diterapkan pada TSP dengan Time Windows (Suprayogi & Mahmudy, 2015), sistem rekomendasi wisata kuliner (Widodo & Mahmudy, 2010), penjadwalan ujian (Mawaddah & Mahmudy, 2006), dan antar jemput laundry (Suprayogi & Mahmudy, 2015). Algoritme ini memanfaatkan teknik pencarian acak auto-adaptif dan membentuk individu baru dengan melakukan *crossover*, mutasi, dan seleksi. Algoritme Genetika mampu menyelesaikan masalah rumit yang sulit untuk diselesaikan dengan menggunakan algoritme tradisional. Dengan memanfaatkan pencarian acak, Algoritme Genetika mampu menjangkau wilayah solusi yang sulit dijangkau oleh algoritme lain, sehingga algoritme ini dirasa cukup baik untuk mengoptimasi algoritme lain untuk memperluas wilayah solusi.

Algoritme Genetika merupakan algoritme yang digunakan dalam melakukan pencarian dengan menggunakan mekanisme genetika dan seleksi alam. Algoritme Genetika lebih mampu menyelesaikan masalah yang kompleks jika dibandingkan dengan metode konvensional (Dasiani dan Arhami, 2006)

Berikut merupakan istilah yang digunakan dalam Algoritme Genetika (Sutojo, M dan Suhartono, 2011):

1. Gen (*genotype*) merupakan variabel yang membentuk sebuah kromosom dengan nilai berupa *integer*, *biner*, *float*, maupun karakter.
2. Kromosom merupakan kumpulan dari gen dengan arti atau fungsi yang sama.
3. Individu merupakan kumpulan kromosom yang memiliki identitas yang sama.
4. Populasi merupakan kelompok individu yang akan melalui proses pada siklus evolusi.
5. Generasi merupakan satuan siklus dalam proses evolusi.
6. Nilai *fitness* merupakan parameter yang menggambarkan tingkat kecocokan hasil dari suatu individu setelah melalui proses evolusi, atau dapat dikatakan sebagai tingkat kecocokan solusi terhadap masalah yang ada.

Algoritme Genetika mampu menyelesaikan masalah dengan sejumlah tahapan sebelum akhirnya mendapatkan solusi yang paling optimum. Calon solusi optimum terlebih dulu direpresentasikan ke dalam bentuk string

kromosom. Setelah melalui beberapa proses yang terdapat di dalam Algoritme Genetika, calon solusi optimum dinilai seberapa baik solusi yang diberikan dengan melakukan fungsi *fitness*. Serangkaian proses Algoritme Genetika yang terus dilakukan dari generasi ke generasi akan menghasilkan kromosom terbaik sebagai sebuah solusi yang diharapkan mendekati optimum dari masalah yang dihadapi (Mahmudy, 2013).

2.9.1 Penentuan Parameter

Parameter kontrol pada Algoritme Genetika merupakan parameter yang harus ditentukan secara tepat agar Algoritme Genetika mampu memberikan nilai yang optimum. Parameter kontrol pada Algoritme Genetika anatara lain ukuran populasi, *crossover rate*, dan *mutation rate*. Penentuan parameter kontrol pada Algoritme Genetika akan mempengaruhi kemampuan Algoritme Genetika dalam melakukan eksplorasi pada calon solusi optimum. Semakin besar nilai pada parameter kontrol yang digunakan, semakin luas area solusi yang mampu dicapai namun disisi lain akan membebani waktu komputasi yang bahkan Algoritme Genetika dapat mengeksplorasi area yang tidak mampu memberikan solusi yang optimum (W. F. Mahmudy, Marian, and Luong 2014).

Kombinasi nilai pada parameter kontrol yang akan digunakan dalam Algoritme Genetika bergantung pada masalah yang akan diselesaikan. Metode untuk menentukan parameter kontrol pada Algoritme Genetika masih belum ditemukan, namun menurut Mahmudy, Marian, dan Luong (2014) untuk mendapatkan parameter kontrol yang paling sesuai perlu dilakukan serangkaian pengujian terhadap parameter kontrol yang akan digunakan agar mendapatkan kombinasi nilai parameter yang paling sesuai. Kusumadewi (2003) dalam bukunya merekomendasikan beberapa kombinasi parameter kontrol yang dapat digunakan sesuai dengan masalah yang ingin diselesaikan, antara lain:

- a. Untuk masalah dengan area solusi yang cukup besar, De Jong merekomendasikan nilai parameter kontrol:

$$(\text{popSize}; \text{cr}; \text{mr}) = (50; 0,6; 0,001)$$

- b. Jika indikator yang digunakan adalah rata-rata *fitness* pada setiap generasi, maka Grefensette merekomendasikan parameter kontrol:

$$(\text{popSize}; \text{cr}; \text{mr}) = (30; 0,95; 0,01)$$

- c. Jika indikator yang digunakan adalah nilai *fitness* dari individu terbaik pada setiap generasi, maka parameter kontrol yang diusulkan adalah:

$$(\text{popSize}; \text{cr}; \text{mr}) = (80; 0,45; 0,01)$$

2.9.2 Representasi Kromosom

Kromosom merupakan sebuah himpunan yang akan merepresentasikan solusi bagi masalah yang akan diselesaikan. Represesntasi kromosom yang tepat sangat mempengaruhi dalam keberhasilan pencarian solusi yang optimal. Setiap individu disusun oleh urutan gen berupa alphabet. Alphabet tersebut dapat

terdiri atas susunan biner (0/1), *floating point*, bilangan *integer*, maupun simbol-simbol, matriks, dan lain-lain (Houck dalam Mahmudy, 2013).

2.9.3 Inisialisasi Populasi

Proses insialisasi populasi pada Algoritme Genetika dilakukan dengan melakukan pembangkitan bilangan acak sesuai dengan representasi kromosom yang telah ditentukan sebelumnya. Ukuran populasi pada Algoritme Genetika akan menentukan jumlah populasi yang dibentuk melalui proses inisialiasi populasi serta populasi yang akan lolos proses seleksi nantinya.

2.9.4 Reproduksi

Reproduksi adalah proses untuk membentuk individu baru dari individu yang terdapat dalam populasi yang telah dibentuk sebelumnya. Proses reproduksi dapat dilakukan dengan proses persilangan (*crossover*) dan atau mutasi. Keturunan yang didapatkan dari hasil reproduksi disebut sebagai *offspring*.

2.9.4.1 Persilangan (*Crossover*)

Pada *crossover*, jumlah *offspring* yang terbentuk bergantung pada nilai *crossover rate* yang telah ditentukan sebelumnya, yang ditunjukkan pada Persamaan 2.14. Pemilihan induk pada *crossover* dilakukan secara acak (Mahmudy, 2013). Pada representasi *real-code*, *offspring* yang akan terbentuk merupakan hasil penghitungan matematis terhadap 2 induk yang telah terpilih seperti pada Persamaan 2.15 dan 2.16.

$$offspring = cRate \times popSize \quad (2.14)$$

$$C_1 = P_1 + \alpha (P_2 - P_1) \quad (2.15)$$

$$C_2 = P_3 + \alpha (P_1 - P_2) \quad (2.16)$$

Keterangan:

P_1 = Parent ke-1

P_2 = Pareng ke-2

C_1 = Child ke-1

C_2 = Child ke-2

α = variabel acak

2.9.4.2 Mutasi

Jumlah *offspring* yang terbentuk pada mutasi bergantung pada nilai *mutation rate* yang telah ditentukan sebelumnya, sesuai Persamaan 2.17. Pada representasi *real-code*, *offspring* yang terbentuk merupakan hasil penghitungan matematis terhadap induk yang telah terpilih seperti pada Persamaan 2.18 .

$$offspring = mRate \times popSize \quad (2.17)$$

$$x'_i = x_i + r (max_i - min_i) \quad (2.18)$$

Keterangan:

x_i' = nilai gen terpilih baru

r = nilai acak antara -0.1 hingga 0.1

\max_i = nilai maximum dari domain gen x_i

\min_i = nilai minimum dari domain gen x_i

2.9.5 Evaluasi

Evaluasi merupakan proses penghitungan nilai *fitness* yang akan merepresentasikan kecocokan solusi pada individu terhadap masalah yang dihadapi. Penentuan nilai *fitness* dapat dilakukan dengan melihat nilai $f(x)$, dalam tahap ini $fitness=f(x)$.

$$fitness = \frac{1}{MAPE} \quad (2.19)$$

2.9.6 Seleksi

Proses seleksi dilakukan untuk menentukan individu yang lolos pada setiap generasi sehingga dapat menjadi induk bagi individu baru selanjutnya. Pada beberapa metode seleksi semakin besar nilai *fitness* pada suatu individu akan membuat individu tersebut memiliki peluang lebih besar untuk lolos pada generasi selanjutnya.

2.9.7 Kriteria Penghentian

Pada Algoritme Genetika terdapat kemungkinan dimana proses pencarian solusi terbaik akan berjalan terus, maka dari itu diperlukan kriteria penghentian dalam proses Algoritme Genetika. Kriteria penghentian yang dapat diterapkan dalam Algoritme Genetika antara lain:

1. Jumlah maksimum generasi tercapai. Jumlah maksimum generasi yang digunakan sebagai kriteria penghenti ditentukan sebelum proses Algoritme Genetika dimulai. Jumlah maksimum generasi yang digunakan ditentukan berdasarkan kompleksitas masalah, umumnya semakin kompleks masalah yang dihadapi semakin besar jumlah generasi yang dibutuhkan. Jumlah maksimum generasi ditentukan sedemikian rupa sehingga tidak ditemukan lagi solusi yang lebih baik jika generasi melebihi jumlah maksimum generasi yang telah ditentukan (Yogeswaran, Ponnambalam & Tiwari dalam Mahmudy 2013).
2. Penghentian iterasi dilakukan jika secara beturut-turut solusi yang didapatkan tidak lebih baik jika dibandingkan solusi sebelumnya (Mahmudy, Marian & Luong 2012).
3. Iterasi dapat dihentikan apabila waktu komputasi telah mencapai waktu maksimum yang telah ditentukan. Kondisi ini dapat diterapkan apabila ingin mengukur performa (W. F. Mahmudy 2013).

Dalam implementasi praktis, dianjurkan untuk menggunakan kombinasi (1) dan (2) (W. Mahmudy, Marian, and Luong 2013).

2.10 Algoritme Genetika – *Backpropagation*

Backpropagation merupakan metode dalam jaringan syaraf tiruan yang mampu melakukan klasifikasi yang baik terhadap masalah dengan cakupan yang cukup luas, namun hal ini terkendala dengan kemungkinan terjadinya konvergensi yang buruk dan tidak tetap. Penerapan *Backpropagation* akan tetap melakukan pelatihan *Backpropagation* untuk mendapatkan bobot yang terbaik pada *local optimum* (Asriningtyas, Dachlan, dan Yudaningtyas, 2015). Langkah-langkah yang dilakukan untuk mendapatkan bobot yang optimum bagi *Backpropagation* diantaranya adalah:

1. Melakukan inisialisasi populasi untuk mendapatkan individu-individu yang merepresentasikan bobot-bobot calon solusi optimum bagi *Backpropagation*.
2. Melakukan *crossover* dan *mutasi* untuk mendapatkan *offspring* yang akan memberikan keturunan yang lebih baik daripada induknya
3. Melakukan pelatihan dengan menggunakan *Backpropagation*.
4. Menghitung nilai *fitness* dari masing-masing individu.
5. Melakukan seleksi untuk menentukan individu-individu yang layak untuk lolos ke generasi selanjutnya.

2.11 Pengukuran Performa

Pengukuran performa dilakukan untuk mengukur seberapa baik suatu algoritme atau sistem yang telah dibuat berjalan. Dalam berbagai penelitian terdapat beberapa fungsi yang digunakan untuk mengukur kinerja suatu sistem, beberapa diantaranya adalah MAPE, MSD (Mean Squared Deviation), MAD (Mean Absolute Deviation) dan sebagainya. Pada penelitian ini alat yang digunakan untuk mengukur kinerja sistem adalah fungsi MAPE (Mean Absolute Percentage Error).

2.11.1 MAPE

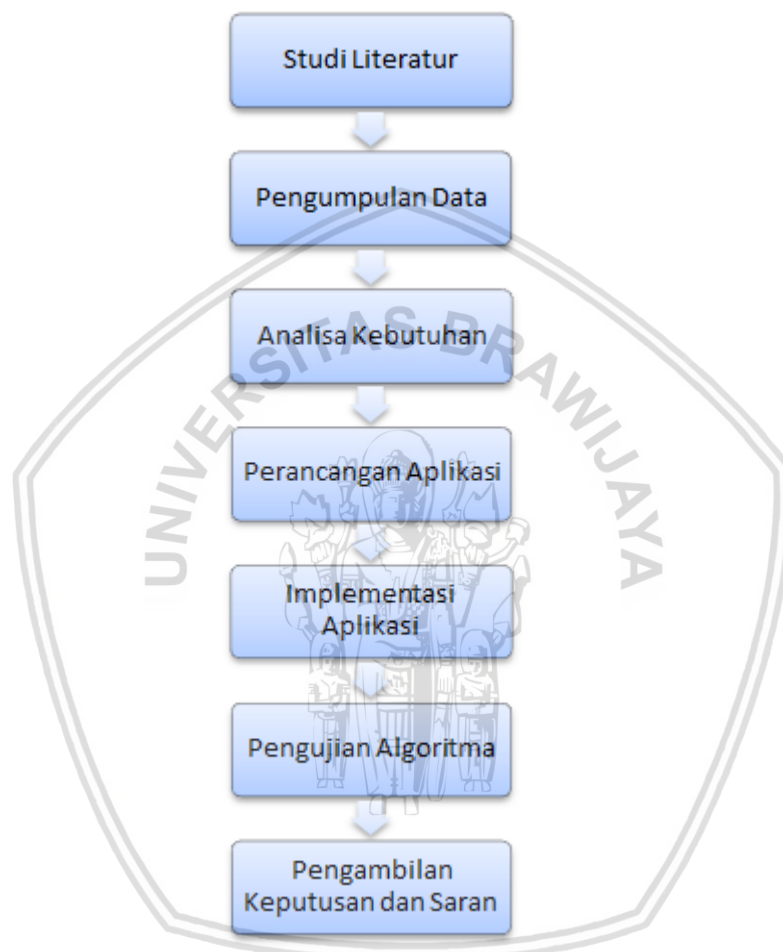
MAPE adalah fungsi yang dapat mengukur rata-rata *error* yang terjadi dalam suatu fungsi *fitness*. Total *error* dapat dilihat dari perbedaan antara hasil optimasi dari GA dan kelas aslinya. Persamaan fungsi MAPE dapat dilihat pada Persamaan 2.11.

$$MAPE = E_i = \frac{1}{n} \sum_{i=1}^n \frac{|P_i - T_i|}{T_i} \times 100 \quad (2.20)$$

Dimana P_i merupakan nilai hasil prediksi dengan menggunakan Algoritme Genetika – *Backpropagation*, T_j merupakan target atau nilai sebenarnya dan n merupakan jumlah data latih yang digunakan dalam penelitian ini. Apabila nilai $P_i = T_j$ maka $E_i = 0$, kondisi ini akan membuat nilai MAPE bergerak dari 0 sampai tak terhingga karena tidak terdapat selisih antar hasil ramalan dan nilai sebenarnya (Tahmasebi & Hezarkhani, 2012).

BAB 3 METODOLOGI

Pada bab 3 akan dijelaskan tahapan-tahapan dalam pelaksanaan penelitian, sehingga tujuan dari penelitian dapat tercapai. Tahapan dapat diilustrasikan seperti pada Gambar 3.1.



Gambar 3.1 Diagram alur metedologi penelitian

3.1 Studi Literatur

Studi literatur adalah tahapan dalam mempelajari penelitian lebih dalam dengan mencari referensi teori yang memiliki kesamaan dan keterkaitan dalam permasalahan yang diteliti. Studi literatur yang dilakukan pada penelitian ini berisi tentang valuta asing, Algoritme Genetika, Jaringan Syaraf Tiruan, *Backpropagation* dan optimasi dengan Algoritme Genetika terhadap *Backpropagation*.

Referensi dalam melakukan pembelajaran didapatkan dari berbagai sumber. Dalam penelitian ini referensi berupa buku, jurnal, laporan penelitian, dan

berbagai sumber lainnya. Tujuan dilakukannya studi literatur adalah sebagai dasar teori dan bagi penulis dalam melakukan penelitian.

3.2 Pengumpulan Data

Pada tahapan ini data mulai dikumpulkan yang nantinya dibutuhkan dalam menyelesaikan permasalahan penelitian. Data yang digunakan dalam penelitian merupakan jenis data sekunder. Data sekunder adalah jenis data yang didapat dari web resmi Bank Indonesia dengan situs yang dapat diakses di www.bi.go.id.

Data yang telah didapatkan, kemudian diolah dan digunakan sebagai parameter input pada proses optimasi menggunakan Algoritme Genetika. Pada penelitian Pramesti, R. A. (2013), Data yang digunakan sebagai parameter input dalam penelitian ini ada 3, yaitu data bulanan dari inflasi, *BI rate*, dan kurs beli bulanan rupiah terhadap dolar Amerika. Semua data tersebut diambil dalam kurun waktu 9 tahun, yaitu dari Januari 2006 hingga Juli 2016 dengan total data 124 data bulanan.

Table 3.1: Analisa kebutuhan dan penelitian.

No.	Kebutuhan Data	Sumber Data	Metode	Kegunaan Data
1	Kurs nilai tengah bulanan	Bank Indonesia	Mengunduh data resmi dari halaman resmi Bank Indonesia www.bi.go.id	Sebagai data latih dan data uji
2	Tingkat suku bunga bulanan (berupa <i>BI rate</i>)	Bank Indonesia	Mengunduh data resmi dari halaman resmi Bank Indonesia www.bi.go.id	Sebagai data latih dan data uji
3.	Tingkat Inflasi bulanan	Bank Indonesia	Mengunduh data resmi dari halaman resmi Bank Indonesia www.bi.go.id	Sebagai data latih dan data uji

3.3 Analisis Kebutuhan

Analisa kebutuhan dilakukan dengan menganalisis dan mengumpulkan informasi mengenai semua kebutuhan yang diperlukan dalam penelitian. Tahapan ini dilakukan agar sistem dapat dibuat dengan benar dan sesuai kebutuhan sehingga dapat menyelesaikan permasalahan dari penelitian. Analisa kebutuhan yang dilakukan berupa identifikasi kebutuhan proses sistem.

3.4 Perancangan

Perancangan aplikasi dibangun dengan berdasarkan hasil pengumpulan data dan analisa kebutuhan yang telah dilakukan. Tahapan ini dilakukan untuk mempermudah pada tahap implementasi dan pengujian.

3.4.1 Deskripsi Sistem

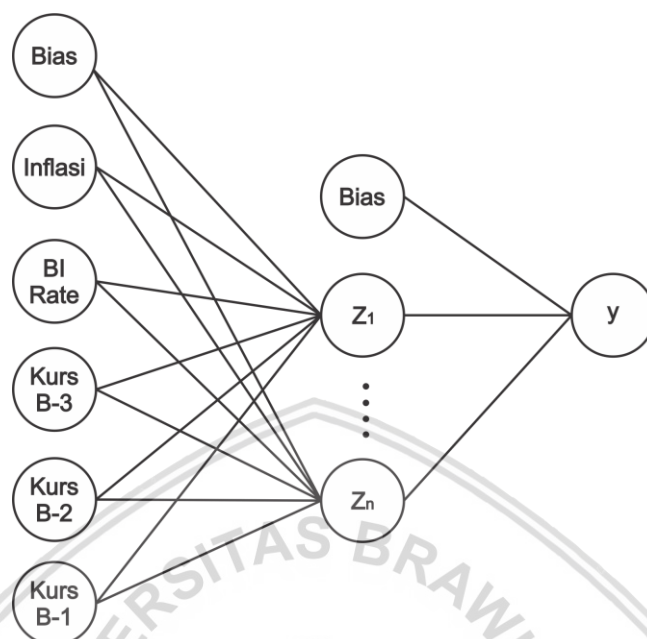
Sistem yang akan dibuat merupakan sistem prediksi nilai tukar rupiah terhadap dolar Amerika dengan menggunakan metode *Backpropagation* yang nilai bobotnya telah melalui tahap optimasi dengan menggunakan Algoritme Genetika. Pada sistem prediksi di tahap pelatihan diperlukan data masukkan berupa *learning rate*, jumlah iterasi, dan jumlah lapisan tersembunyi pada arsitektur yang akan digunakan pada *Backpropagation*, serta jumlah generasi, ukuran populasi, *crossover rate*, dan *mutation rate* pada Algoritme Genetika. Sedangkan pada tahap uji diperlukan data masukkan berupa nilai inflasi, nilai *BI rate*, nilai kurs 3 bulan terakhir, nilai kurs 2 bulan terakhir, dan nilai kurs 1 bulan terakhir Implementasi.

Implementasi aplikasi prediksi nilai tukar valuta asing dengan menggunakan java, database mysql dan perangkat lunak pendukung lainnya. Adapun implementasi sistem adalah sebagai berikut:

- Pembuatan antarmuka yang akan dibuat dengan menggunakan bahasa pemrograman Java.
- Menerapkan metode Algoritme Genetika dalam proses pelatihan *Backpropagation* untuk mendapatkan bobot terbaik.
- Setelah mendapatkan bobot terbaik, bobot akan dilatih menggunakan algoritme *Backpropagation*.
- Bobot terbaik hasil pelatihan digunakan untuk melakukan prediksi dengan menggunakan algoritme *feedforward* pada algoritme *Backpropagation*.
- Kemudian akan didapatkan hasil prediksi nilai tukar valuta asing pada bulan selanjutnya dan dihitung besar nilai kesalahannya.

3.4.2 Arsitektur Backpropagation

Pada penelitian ini arsitektur yang akan digunakan merupakan *Backpropagation* dengan 5 masukan dan 1 bias pada lapisan masukan dan 1 bias pada lapisan tersembunyi serta 1 keluaran, sedangkan jumlah *neuron* pada lapisan tersembunyi akan diujikan. Gambar 3.2 menunjukkan arsitektur *Backpropagation* yang menggunakan 2 *neuron* pada lapisan tersembunyi.



Gambar 3.2 *Arsitektur Backpropagation*

Masukan yang digunakan pada arsitektur *Backpropagation* pada penelitian adalah sebanyak 5 dengan masing-masing masukan mewakili nilai inflasi bulanan, tingkat suku bunga bulanan (*BI rate*), kurs pada 3 bulan sebelumnya, 2 bulan sebelumnya dan sebulan sebelumnya serta bias pada lapisan masukan dan lapisan tersembunyi yang akan selalu bernilai 1.

3.5 Pengujian Algoritme

Tahapan pengujian bertujuan untuk melakukan pengujian terhadap algoritme yang telah dibuat. Sehingga dari hasil pengujian akan diketahui apakah algoritme yang telah dibuat sesuai dengan yang diharapkan atau tidak. Pengujian dilakukan dengan memasukkan nilai parameter *Backpropagation* dan Algoritme Genetika yang berbeda terhadap data latih. Tujuannya yaitu untuk mengetahui pengaruh parameter yang digunakan pada proses pelatihan. Setelah didapatkan parameter yang memberi bobot terbaik kemudian dilakukan proses prediksi. Dan setelah peramalan selesai dilakukan, maka bobot terbaik yang dihasilkan akan dievaluasi menggunakan sejumlah data uji, kemudian akan dihitung tingkat kesalahan peramalan.

3.6 Pengambilan Kesimpulan dan Saran

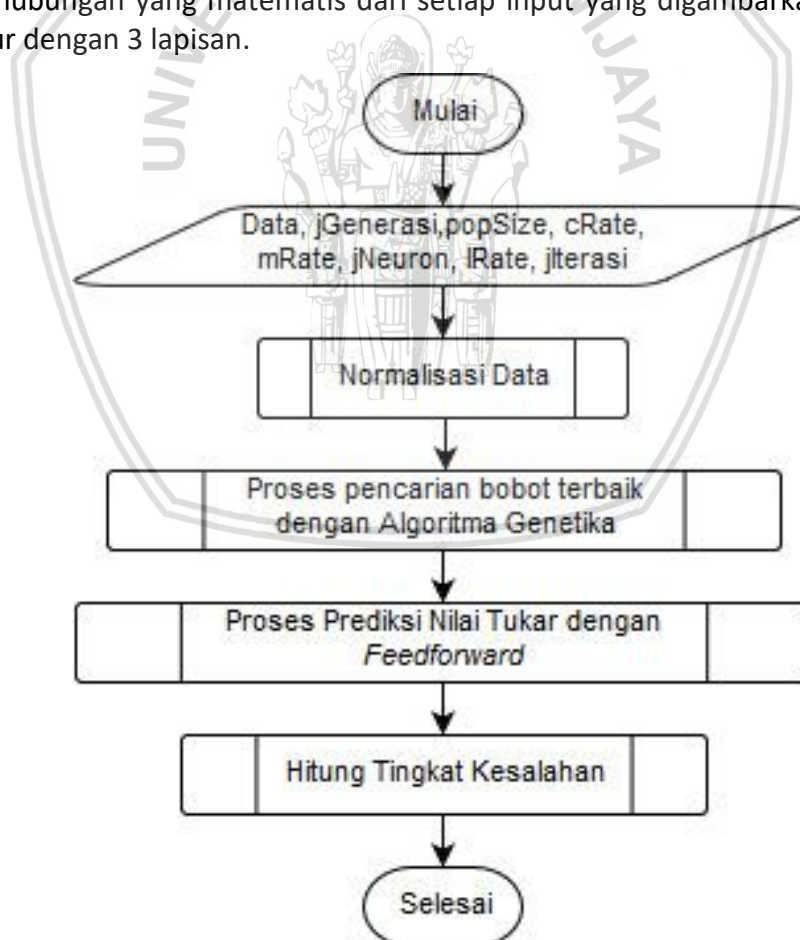
Pengambilan kesimpulan dilakukan setelah semua proses mulai dari studi literatur hingga pengujian telah selesai dilakukan. Tahapan pengambilan kesimpulan dilakukan untuk mengetahui efektifitas kinerja dari algoritme yang telah digunakan. Setelah itu tahapan akhir yang dilakukan adalah penulisan saran yang dapat membantu dalam pengembangan algoritma selanjutnya.

BAB 4 ALGORITME DAN PERANCANGAN

Perancangan sebelum mengimplementasikan atau membuat suatu sistem diperlukan agar seluruh kebutuhan sistem telah terpenuhi. Selain itu perancangan dapat membantu sistem dalam menghemat waktu pengerjaan karena segala kebutuhan sistem tersedia. Pada bab ini dijelaskan tentang penyelesaian masalah dengan tahapan yang telah ditentukan serta rancangan atarmuka sistem yang akan dibangun.

4.1 Formulasi Permasalahan

Prediksi nilai tukar rupiah terhadap dolar Amerika menggunakan data-data historis dari faktor-faktor yang mempengaruhi perubahan pada nilai. Faktor-faktor yang mempengaruhi nilai tukar rupiah antara lain adalah inflasi, *BI rate* dan data nilai tukar rupiah terhadap dolar Amerika 3 bulan terakhir. Faktor-faktor tersebut akan dijadikan input pada *Backpropagation* yang kemudian dicari bobot optimumnya menggunakan Algoritme Genetika. Algoritme Genetika akan mencari bobot yang optimum bagi *Backpropagation* untuk mampu membentuk sebuah hubungan yang matematis dari setiap input yang digambarkan kedalam arsitektur dengan 3 lapisan.



Gambar 4.1 Diagram alur sistem

Dalam melakukan proses prediksi nilai tukar rupiah terhadap dolar Amerika menggunakan *Backpropagation* dengan melakukan pembobotan pada Algoritme Genetika akan dijelaskan pada Gambar 4.1. Pada diagram alur pada Gambar 4.1 proses diawali dengan memberikan masukan berupa data yang akan digunakan sebagai data latih (*Data*), jumlah generasi yang harus dicapai Algoritme Genetika untuk berhenti (*jGenerasi*), ukuran populasi pada Algoritme Genetika (*popSize*), nilai probabilitas *crossover* (*cRate*), nilai probabilitas mutasi (*mRate*), jumlah *neuron* pada lapisan tersembunyi (*jNeuron*), nilai *learning rate* (*lRate*) untuk mengatur laju pembelajaran pada proses pelatihan *Backpropagation* di dalam Algoritme Genetika, serta jumlah iterasi (*jIterasi*) yang dilakukan pada pelatihan *Backpropagation* di dalam Algoritme Genetika. Setelah semua parameter terpenuhi lakukan proses normalisasi pada data latih dan data uji, lakukan proses pelatihan dengan Algoritme Genetika – *Backpropagation*. Setelah mendapatkan bobot terbaik dari pelatihan, lakukan pengujian terhadap data uji dan diikuti dengan penghitungan tingkat kesalahan. Lebih rinci alur sistem dijelaskan dalam bentuk contoh kasus berikut.

Bank Indonesia ingin melakukan prediksi terhadap nilai tukar rupiah terhadap dolar Amerika dengan menggunakan metode *Backpropagation* yang melakukan pembobotan dengan Algoritme Genetika. Langkah-langkah dalam melakukan prediksi antara lain:

1. Tahap pertama adalah mengumpulkan data inflasi, *BI rate*, dan perubahan nilai tukar rupiah terhadap dolar Amerika pada 3 bulan sebelumnya. Data terdiri atas 99 data latih dan 2 data uji. Data latih merupakan data yang akan digunakan dalam pelatihan Algoritme Genetika untuk mendapatkan bobot yang paling optimum. Sedangkan data latih merupakan data yang digunakan dalam pengujian untuk mendapatkan besar *error* yang dihasilkan dari bobot yang telah didapat.
2. Data yang telah didapat kemudian dilakukan normalisasi. Normalisasi yang digunakan merupakan normalisasi min max. Normalisasi perlu dilakukan untuk mendapatkan nilai yang sesuai dengan domain fungsi aktivasi *sigmoid* yaitu pada range 0 hingga 1. Namun karena pada fungsi aktivasi *sigmoid* tidak pernah tercapai maka pada normalisasi ini digunakan nilai minimum 0.1 dan nilai maksimum 0.9.
3. Masuk pada proses Algoritme Genetika, yaitu penentuan individu awal secara acak sejumlah populasi yang telah ditentukan dengan representasi kromosom sesuai dengan Gambar 4.3.
4. Individu yang telah terbentuk dilakukan reproduksi berupa *crossover* antara dua induk dan juga mutasi.
5. Hitung nilai *fitness* setiap individu beserta *offspring* yang telah didapatkan dari hasil reproduksi dengan melakukan pelatihan melalui proses *Backpropagation*. Nilai *fitness* didapat dari nilai *error* individu yang telah dilatih terhadap data latih. Individu akan berhenti melakukan proses

- pelatihan setelah pelatihan melebihi jumlah iterasi atau jika selama lima iterasi berturut-turut nilai *fitness* semakin memburuk.
6. Setiap individu kemudian akan diseleksi. Proses seleksi yang digunakan dalam masalah ini adalah *Roulette Wheel* dimana setiap individu akan dicari nilai probabilitas kumulatifnya berdasarkan nilai *fitness*nya.
 7. Proses Algoritme Genetika akan terus dilakukan hingga generasi yang telah ditentukan.
 8. Bobot terbaik hasil pelatihan menggunakan Algoritme Genetika akan digunakan sebagai bobot pada proses pengujian dengan *feedforward*.
 9. Hasil pengujian dan nilai *error* dapat dilihat sebagai tolak ukur keakuratan prediksi nilai tukar Rupiah terhadap dolar Amerika.

Tahap-tahap tersebut secara lebih ringkas dijelaskan melalui diagram alur sistem sesuai pada Gambar 4.2.

4.2 Normalisasi Min Max

Data inflasi, *BI rate*, dan perubahan nilai tukar rupiah terhadap dolar Amerika pada 3 bulan sebelumnya yang telah dikumpulkan sebelumnya kemudian perlu dilakukan Normalisasi. Data yang didapatkan terdiri atas data latih yang contohnya terdapat pada Tabel 4.1 dan data uji yang contohnya terdapat pada Tabel 4.2. Contoh data latih yang digunakan untuk mendapatkan bobot optimal pada *Backpropagation* akan ditunjukkan pada Tabel 4.1.

Tabel 4.1 Contoh data latih permasalahan

Inflasi	<i>BI rate</i>	Kurs 3 Bulan Sebelumnya	Kurs 2 Bulan Sebelumnya	Kurs 1 Bulan Sebelumnya	Perubahan Kurs Bulan ini
0.1574	0.1275	9540.4	9299.35	9217.476	8981.667
0.0241	0.0575	12009.1	12112.03	11708.83	9117.849
0.0489	0.075	14468	13864.76	13740.95	13923.75
0.0442	0.07	13923.75	13958.35	13583.2	13259.14
0.0683	0.075	13441.79	13850.7	14468	13864.76
0.0335	0.075	13864.76	13740.95	13923.75	13958.35
0.0257	0.065	10027.5	9950.389	9530	9517.2
0.0547	0.0575	9758.105	9772.955	9809.909	9931
0.0461	0.0675	8576.19	8574.789	8809.45	8939.667
0.0461	0.0675	8598.8	8607	8576.19	8574.789

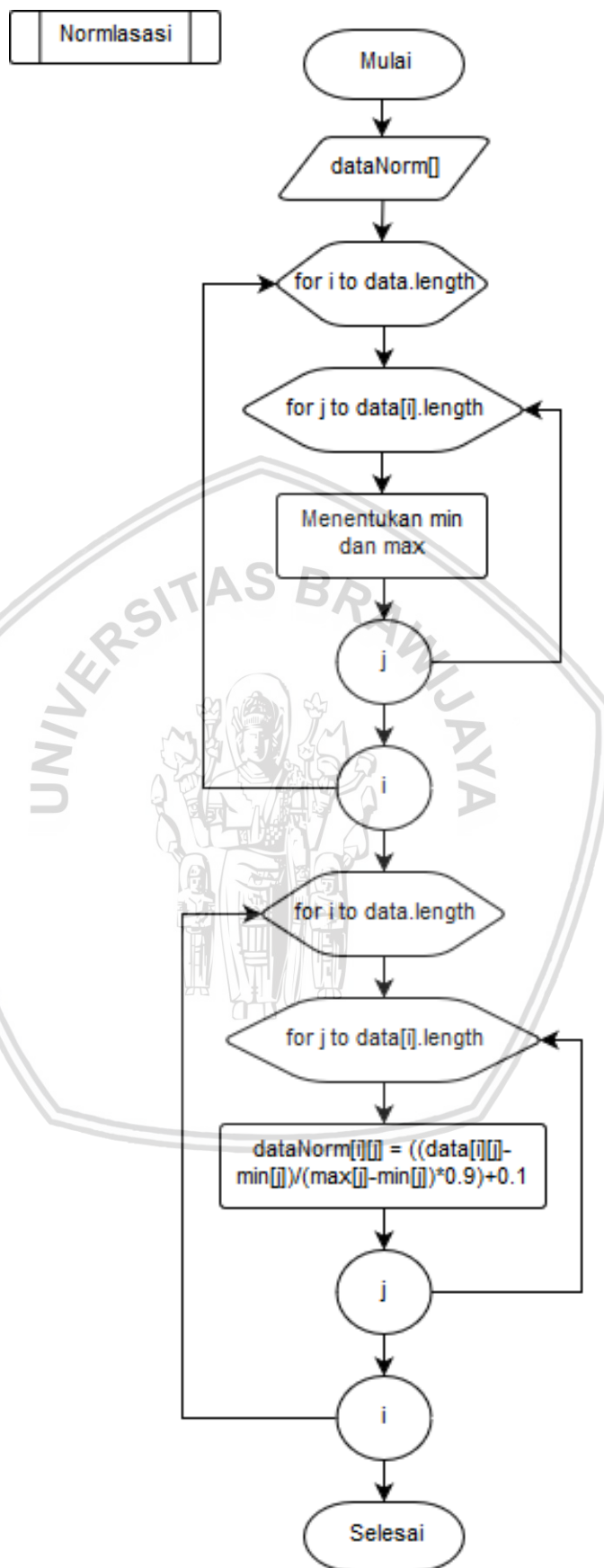
Contoh data uji untuk bulan yang akan diprediksi oleh Bank Indonesia ditunjukkan pada Tabel 4.2.

Tabel 4.2 Contoh data uji permasalahan

Inflasi	BI rate	Kurs 3 Bulan Sebelumnya	Kurs 2 Bulan Sebelumnya	Kurs 1 Bulan Sebelumnya	Kurs Bulan ini
0.156	0.125	9217.476	8981.667	9029.857	9409.636
0.154	0.1275	9299.35	9217.476	8981.667	9029.857
0.0445	0.0675	13958.35	13583.2	13259.14	13245.81
0.0625	0.075	13850.7	14468	13864.76	13740.95
0.0414	0.0725	13740.95	13923.75	13958.35	13583.2
0.0718	0.075	13379.95	13441.79	13850.7	14468
0.0241	0.065	9950.389	9530	9517.2	9504.85
0.0557	0.0575	9735.05	9758.105	9772.955	9809.909
0.0442	0.065	8574.789	8809.45	8939.667	9060.227
0.0479	0.0675	8607	8576.19	8574.789	8809.45

Data pada Tabel 4.1 dan Tabel 4.2 kemudian dilakukan normalisasi sehingga menghasilkan data pada Tabel 4.3 dan Tabel 4.4. Normalisasi yang digunakan merupakan normalisasi min max. Normalisasi perlu dilakukan untuk mendapatkan nilai yang sesuai dengan domain fungsi aktivasi *sigmoid* yaitu pada range 0 hingga 1. Namun karena pada fungsi aktivasi *sigmoid* tidak pernah tercapai maka pada normalisasi ini digunakan nilai minimum 0.1 dan nilai maksimum 0.9. Proses normalisasi ditunjukkan sesuai dengan diagram alur pada Gambar 4.2.

Pada Gambar 4.2 dijelaskan alur normalisasi yang diawali dengan melakukan inisialisasi array `dataNorm` yang akan menyimpan data hasil normalisasi. Kemudian melakukan pengulangan sebanyak jumlah data dan sebanyak parameter masukan untuk mendapatkan nilai maksimum dan minimum dari setiap parameter masukan dari semua data. Kemudian lakukan pengulangan lagi pada setiap data pada setiap parameter dan lakukan normalisasi pada data sesuai dengan Persamaan 2.1



Gambar 4.2 Diagram alur normalisasi

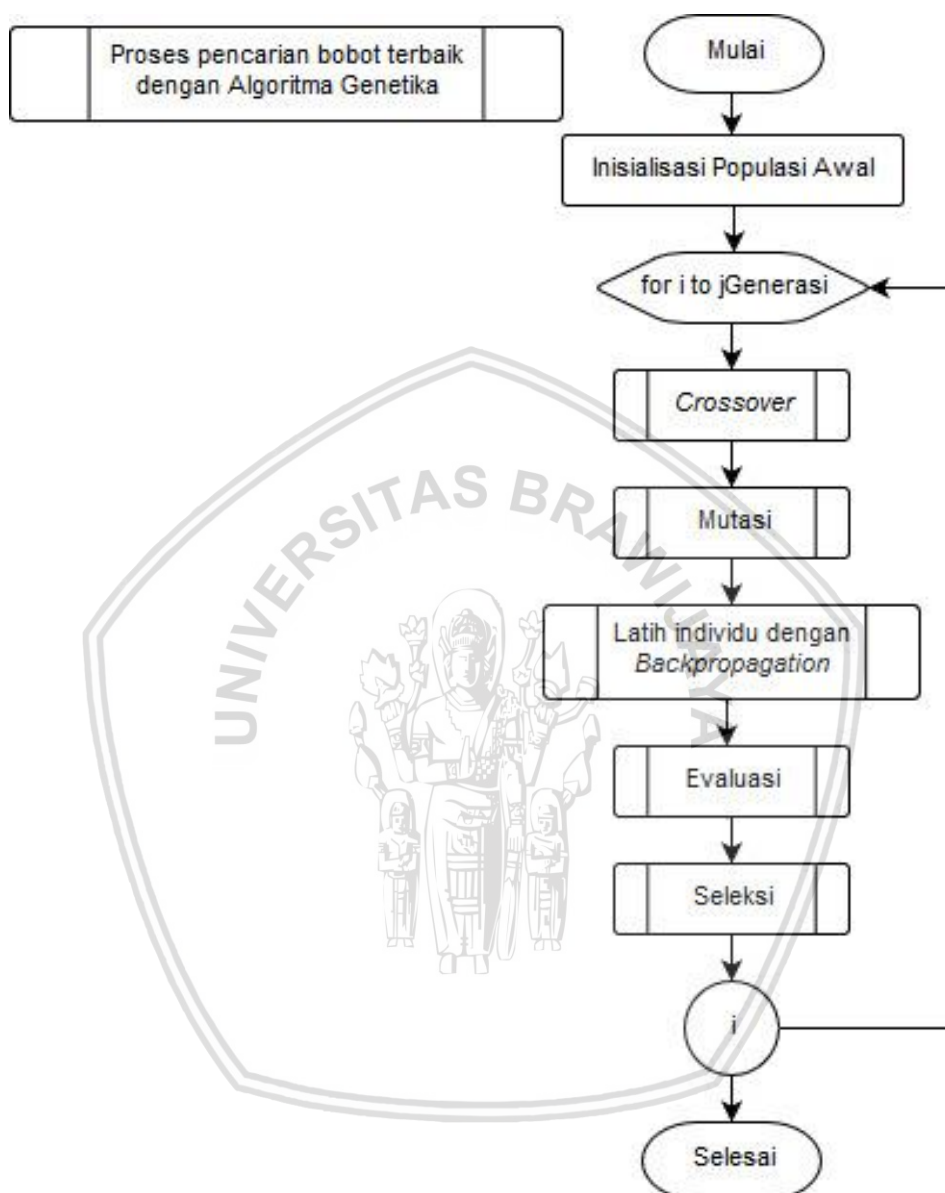
Tabel 4.3 Hasil normalisasi contoh data latih

Inflasi	BI rate	Perubahan Kurs 3 Bulan Sebelumnya	Perubahan Kurs 2 Bulan Sebelumnya	Perubahan Kurs 1 Bulan Sebelumnya	Perubahan Kurs Bulan ini
0.9	0.9	0.231081	0.198359	0.187244	0.155233
0.887397	0.871429	0.155233	0.161775	0.21333	0.180935
0.248837	0.3	0.9	0.818111	0.801304	0.826118
0.22063	0.242857	0.826118	0.830815	0.779889	0.735898
0.365266	0.3	0.760693	0.816202	0.9	0.818111
0.156414	0.3	0.818111	0.801304	0.826118	0.830815
0.109602	0.185714	0.297205	0.286737	0.229669	0.227932
0.283646	0.1	0.260634	0.26265	0.267667	0.284105
0.232033	0.214286	0.10019	0.1	0.131855	0.149532
0.232033	0.214286	0.103259	0.104373	0.10019	0.1

Tabel 4.4 Hasil normalisasi contoh data uji

Inflasi	BI rate	Perubahan Kurs 3 Bulan Sebelumnya	Perubahan Kurs 2 Bulan Sebelumnya	Perubahan Kurs 1 Bulan Sebelumnya	Perubahan Kurs Bulan ini
0,891598	0,871429	0,187244	0,155233	0,161775	0,21333
0,879595	0,9	0,198359	0,187244	0,155233	0,161775
0,222431	0,214286	0,830815	0,779889	0,735898	0,734088
0,330458	0,3	0,816202	0,9	0,818111	0,801304
0,203826	0,271429	0,801304	0,826118	0,830815	0,779889
0,386272	0,3	0,752298	0,760693	0,816202	0,9
0,1	0,185714	0,286737	0,229669	0,227932	0,226255
0,289647	0,1	0,257505	0,260634	0,26265	0,267667
0,22063	0,185714	0,1	0,131855	0,149532	0,165898
0,242836	0,214286	0,104373	0,10019	0,1	0,131855

4.3 Siklus Penentuan Bobot Optimum *Backpropagation* dengan Algoritme Genetika



Gambar 4.3 Diagram alur Algoritme Genetika dan *Backpropagation*

Tahapan yang dilakukan dalam melakukan prediksi dengan menggunakan Algoritme Genetika secara anatra lain:

1. Inisialisasi populasi awal
2. Reproduksi
3. Latih setiap individu dengan *Backpropagation*
4. Evaluasi/penghitungan *fitness* dengan menggunakan *feedforward*
5. Seleksi

Langkah-langkah dalam pelatihan menggunakan Algoritma Genetika untuk mendapatkan bobot *Backpropagation* terbaik untuk melakukan prediksi nilai tukar rupiah terhadap dolar Amerika dengan lebih jelas akan digambarkan dalam bentuk diagram alur seperti Gambar 4.3.

Pada Gambar 4.3 menunjukkan alur proses yang dilakukan untuk mendapatkan bobot yang paling optimal untuk *Backpropagation* dengan menggunakan Algoritma Genetika. Pada proses ini perlu dilakukan inisialisasi populasi, inisialisasi individu dilakukan sebanyak jumlah populasi yang telah ditentukan. Pada setiap individu representasi kromosom disesuaikan dengan arsitektur jumlah neuron pada lapisan tersembunyi *Backpropagation*. Setelah semua individu terbentuk selanjutnya melakukan proses *crossover* dan mutasi. Metod *crossover* yang digunakan merupakan *extended intermediate crossover*, dimana *offspring* yang dihasilkan merupakan hasil kombinasi dari kedua induk. Mutasi dilakukan dengan menggunakan metode *random mutation*, metode ini dilakukan pada satu induk terpilih dengan menambahkan atau mengurangi satu gen terpilih dengan nilai yang relatif kecil. Setelah melakukan proses reproduksi, kemudian semua individu dan *offspring* dilatih menggunakan *Backpropagation*. Pelatihan dilakukan pada setiap individu dan *offspring* hanya hingga individu dan *offspring* mencapai *local optimum*, yaitu titik dimana saat individu dan *offspring* sudah memberikan nilai *error* terkecil yang bila dilanjutkan akan menyebabkan nilai *error* membesar. Individu dan *offspring* yang telah dilatih kemudian dihitung nilai *fitnessnya* dengan rumus seperti pada Persamaan 2.19. Kemudian setiap individu dan *offspring* diseleksi untuk lanjut sebagai individu baru pada generasi selanjutnya. Pada kasus ini seleksi yang digunakan merupakan *Roulette Wheel*. Proses ini dilakukan hingga mencapai jumlah generasi (*jGenerasi*) yang telah ditentukan.

4.3.1 Representasi Kromosom

Pada tahap representasi kromosom pada permasalahan prediksi nilai tukar rupiah terhadap dolar Amerika kromosom akan merepresentasikan bobot-bobot dari *Backpropagation* dimana bobot tersebut berupa bilangan real sehingga dalam representasi kromosomnya menggunakan pengkodean real (*real-code*). Pada Gambar 4.2 menunjukkan representasi kromosom dengan pengkodean real pada *Backpropagation* yang memiliki 2 neuron pada layer tersembunyi.

V01	V11	V21	V31	V41	V51	V02	V12	V22
V32	V42	V52	W01	W11	W21	Fitness		

Gambar 4.4 Representasi kromosom *real-code* dengan 2 neuron pada lapisan tersembunyi

Keterangan:

1. V01, V11, V21, V31, V41, dan V51 merupakan segmen yang merepresentasikan bobot dari lapisan masukan ke lapisan tersembunyi. Sedangkan W01, W11, dan W21 merupakan segmen yang merepresentasikan bobot dari lapisan tersembunyi ke lapisan keluaran.
2. V11 dan V12 merupakan segmen yang merepresentasikan bobot dari masukan inflasi ke lapisan tersembunyi.
3. V21 dan V22 merupakan segmen yang merepresentasikan bobot dari masukan Tingkat Suku Bunga (*BI rate*) ke lapisan tersembunyi.
4. V31 dan V33 merupakan segmen yang merepresentasikan bobot dari masukan selisih kurs 3 bulan sebelumnya ke lapisan tersembunyi.
5. V41 dan V44 merupakan segmen yang merepresentasikan bobot dari masukan selisih kurs 2 bulan sebelumnya ke lapisan tersembunyi.
6. V51 dan V52 merupakan segmen yang merepresentasikan bobot dari masukan selisih kurs 1 bulan sebelumnya ke lapisan tersembunyi.
7. V01 merupakan segmen yang merepresentasikan bobot dari masukan bias ke lapisan tersembunyi.

4.3.2 Inisialisasi Populasi Awal

Langkah-langkah untuk menentukan populasi awal pada Algoritme Genetika adalah sebagai berikut:

1. Menentukan jumlah populasi dan nilai yang dibangkitkan secara random untuk representasi kromosom pada populasi awal.
2. Proses pembentukan kromosom akan dilakukan dengan angka random yang telah didapatkan.
3. Apabila jumlah populasi awal yang telah ditentukan terpenuhi maka akan terbentuk populasi awal, jika belum terpenuhi maka pembentukan kromosom akan dilanjutkan.

Hasil pembangkitan secara acak akan dijelaskan pada Tabel 4.5. Pembangkitan nilai acak pada populasi awal dilakukan dengan batas-batas tertentu, dimana gen pada tiap individu diberikan nilai acak antara 0 hingga 1.

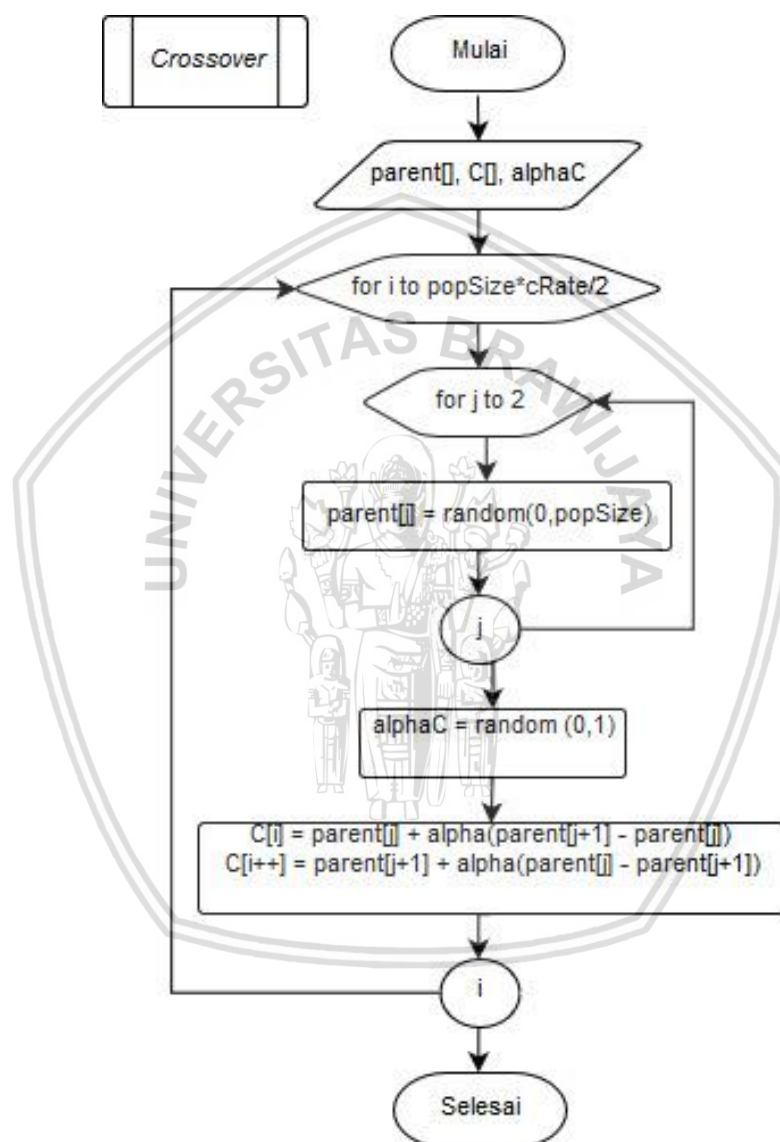
Tabel 4.5 Individu awal

Individu	v01	v11	v21	v31	v41	v51	v02	v12	v22	v32	v42	v52	w01	w11	w21
<i>Parent 1</i>	0,7	0,2	0,5	0,7	0,1	0,83	0,87	0,36	0,84	0,36	0,37	0,38	0,57	0,91	0,94
<i>Parent 2</i>	0,64	0,69	0,81	0,12	0,46	0,63	0,31	0,63	0,71	0,19	0,29	0,64	0,88	0,91	0,67
<i>Parent 3</i>	0,54	0,13	0,64	0,41	0,71	0,64	0,65	0,46	0,19	0,76	0,18	0,82	0,84	0,65	0,64
<i>Parent 4</i>	0,64	0,17	0,92	0,74	0,77	0,53	0,74	0,71	0,91	0,16	0,28	0,4	0,47	0,74	0,73
<i>Parent 5</i>	0,75	0,38	0,48	0,29	0,59	0,933	0,83	0,65	0,64	0,36	0,59	0,65	0,55	0,91	0,95
<i>Parent 6</i>	0,85	0,15	0,7	0,16	0,57	0,4	0,45	0,75	0,19	0,58	0,94	0,49	0,02	0,36	0,12
<i>Parent 7</i>	0,74	0,94	0,99	0,65	0,46	0,75	0,19	0,14	0,75	0,74	0,56	0,18	0,85	0,95	0,46
<i>Parent 8</i>	0,65	0,15	0,31	0,47	0,31	0,86	0,97	0,88	0,56	0,47	0,49	0,85	0,43	0,95	0,73
<i>Parent 9</i>	0,75	0,45	0,65	0,75	0,65	0,89	0,97	0,75	0,47	0,32	0,75	0,31	0,19	0,75	0,67
<i>Parent 10</i>	0,56	0,51	0,75	0,86	0,78	0,51	0,85	0,89	0,89	0,17	0,14	0,75	0,15	0,87	0,94

4.3.3 Reproduksi

Reproduksi merupakan proses pembentukan individu baru dengan memanfaatkan individu yang ada sebagai induk. Pada proses reproduksi terdapat 2 proses lain didalamnya, yaitu *crossover* dan mutasi.

4.3.3.1 Crossover



Gambar 4.5 Diagram alur *crossover*

Crossover merupakan proses persilangan antara dua induk terpilih. Secara singkat tahapan *crossover* dijelaskan pada Gambar 4.5. Pada Gambar 4.5 proses diawali dengan melakukan inisialisasi array *parent* yang nantinya berisikan induk terpilih, inisialisasi array *C* yang akan berisi *offspring-offspring* yang akan terbentuk dari hasil *crossover*, dan *alphaC* yang merupakan nilai random.

Crossover akan menghasilkan jumlah *offspring* sesuai dengan nilai dari hasil kali jumlah populasi (*popSize*) dengan peluang *crossover* (*cRate*) seperti pada Persamaan 2.14. Dengan perhitungan seperti berikut.

$$cRate = 0,2$$

$$offspring = popSize \times cRate$$

$$offspring = 10 \times 0,2$$

$$offspring = 2$$

Pemilihan induk pada proses *crossover* dilakukan secara acak. *Crossover* dilakukan dengan menambahkan atau mengurangi nilai gen induk terhadap selisih gen kedua induk yang dikali dengan nilai *alpha*. Jika jumlah *offspring* yang dihasilkan kurang dari proses *crossover* belum memenuhi jumlah *offspring* maka proses akan dilakukan lagi, namun jika sudah proses dihentikan. Proses pembentukan *offspring* dilakukan sesuai dengan Persamaan 2.15 dan Persamaan 2.16. Berikut merupakan penghitungan dalam melakukan *crossover*.

Misalkan induk yang didapat dari hasil random merupakan *Parent 4* dan *Parent 9*. *Crossover* akan menghasilkan 2 *offspring*, yaitu *Child 1* dan *Child 2*.

P4

Individu	V10	V11	V12	V13	V14	V15	V20	V21	V22
Parent 4	0,64	0,17	0,92	0,74	0,77	0,53	0,74	0,71	0,91
V23	V24	V25	W10	W11	W12				
0,16	0,28	0,4	0,47	0,74	0,73				

Gambar 4.6 Kromosom dari *Parent 4*

P9

Individu	V10	V11	V12	V13	V14	V15	V20	V21	V22
Parent 9	0,75	0,45	0,65	0,75	0,65	0,89	0,97	0,75	0,47
V23	V24	V25	W10	W11	W12				
0,32	0,75	0,31	0,19	0,75	0,67				

Gambar 4.7 Kromosom dari *Parent 9*

Misalkan angka random untuk α yang terbentuk seperti pada Gambar 4.8.

alpha	V10	V11	V12	V13	V14	V15	V20	V21	V22
	-0,4	0,21	0,43	-0,18	-0,14	0,74	0,25	-0,77	0,56
V23	V24	V25	W10	W11	W12				
0,32	0,28	0,91	-0,32	-0,56	0,43				

Gambar 4.8 Alpha untuk *crossover* antara *Parent 4* dan *Parent 9*

Maka penghitungan sesuai dengan Persamaan 2.15 dan Persamaan 2.16 adalah

$$\begin{aligned} Child1 &= 0,64 - 0,4 (0,75 - 0,64) &= 0,596 \\ &= 0,64 - 0,044 \end{aligned}$$

$$\begin{aligned} \text{Child 2} &= 0,65 - 0,56 (0,77 - 0,65) \\ &= 0,65 - 0,0672 \end{aligned}$$

Contoh penghitungan diatas dilakukan pada gen pertama, penghitungan tersebut terus dilanjutkan hingga gen terakhir dengan nilai *alpha* menyesuaikan dengan urutan gen yang dihitung sehingga akhirnya menghasilkan *Child 1* dan *Child 2* seperti pada Gambar 4.9 dan Gambar 4.10.

Child 1

Individu	V10	V11	V12	V13	V14	V15	V20	V21	V22
<i>Child 1</i>	0,596	0,2288	0,8039	0,7382	0,7868	0,7964	0,7975	0,6792	0,6636
V23	V24	V25	W10	W11	W12				
0,2112	0,4116	0,3181	0,5596	0,7344	0,7042				

Gambar 4.9 Kromosom dari *Child 1* hasil *crossover*

Child 2

Individu	V10	V11	V12	V13	V14	V15	V20	V21	V22
<i>Child 2</i>	0,794	0,3912	0,7661	0,7518	0,6332	0,6236	0,9125	0,7808	0,7164
V23	V24	V25	W10	W11	W12				
0,2688	0,6184	0,3919	0,1004	0,7556	0,6958				

Gambar 4.10 Kromosom dari *Child 2* hasil *crossover*

4.3.3.2 Mutasi

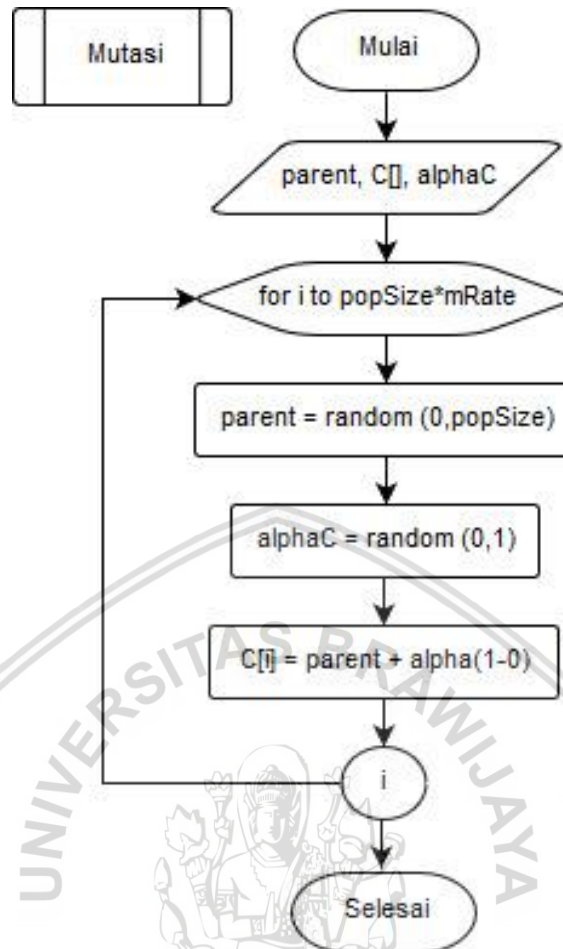
Mutasi dilakukan dengan memilih 1 induk secara acak dari populasi yang ada. Mutasi menggunakan metode dimana nilai gen awal ditambahkan atau dikurangkan dengan hasil perkalian gen dengan nilai random. Secara singkat tahapan dalam mutasi dijelaskan pada Gambar 4.11. Mutasi dimulai dengan melakukan inisialisasi variable *parent* untuk menampung calon induk terpilih, inisialisasi *alphaC* sebagai nilai acak pada proses pembentukan *offspring*, inisialisasi array C yang akan mampung calon *offspring* terbentuk. Proses mutasi dilakukan hingga jumlah tertentu. *Offspring* yang terbentuk dalam proses mutasi akan bergantung pada ukuran populasi dan nilai *mutation rate* sesuai dengan persamaan 2.17 dengan penghitungan seperti berikut.

$$mRate = 0,1$$

$$offspring = popSize \times mRate$$

$$offspring = 10 \times 0,1$$

$$offspring = 1$$



Gambar 4.11 Diagram alur mutasi

Misalkan mutasi dilakukan pada *Parent 8* dengan gen yang terpilih adalah gen ke-8. Mutasi akan menghasilkan 1 *offspring*, yaitu *Child 3*.

Parent 8

Parent	V10	V11	V12	V13	V14	V15	V20	V21	V22
<i>Parent 8</i>	0,65	0,15	0,31	0,47	0,31	0,86	0,97	0,88	0,56
V23	V24	V25	W10	W11	W12				
0,47	0,49	0,85	0,43	0,95	0,73				

Gambar 4.12 Kromosom dari *Parent 8*

Misalkan angka random untuk α yang terpilih adalah 0,31 maka penghitungan sesuai dengan Persamaan 2.18 adalah

$$\begin{aligned}
 \text{Child 3} &= 0,88 + 0,31 (1-0) \\
 &= 0,88 + 0,31 \\
 &= 1,19
 \end{aligned}$$

Child 3

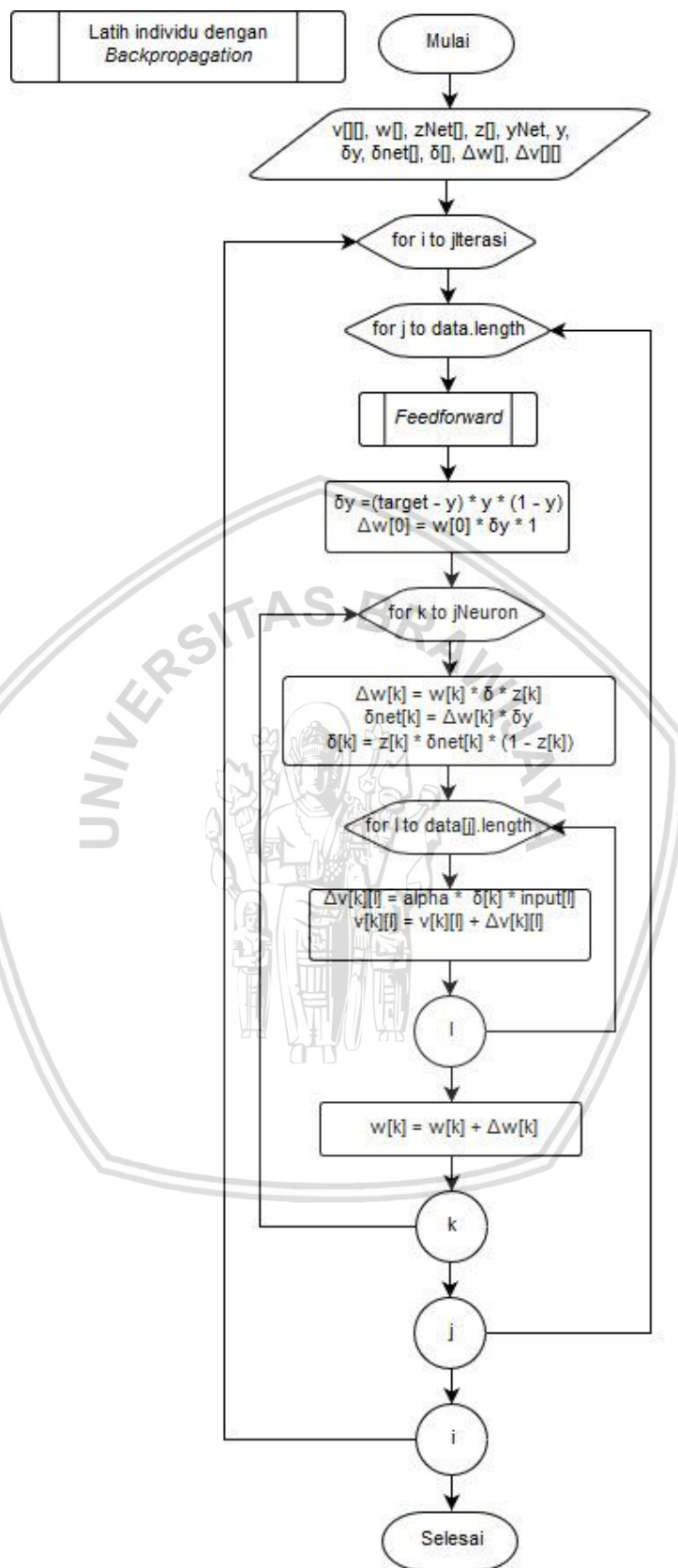
Parent	V10	V11	V12	V13	V14	V15	V20	V21	V22
Child 3	0,65	0,15	0,31	0,47	0,31	0,86	0,97	1,19	0,56
V23	V24	V25	W10	W11	W12				
0,47	0,49	0,85	0,43	0,95	0,73				

Gambar 4.13 Kromosom dari Child 3 hasil mutasi

Dari kedua proses tersebut didapatkan 3 *offspring* yaitu *Child 1*, *Child 2*, dan *Child 3* yang kemudian akan dilakukan proses pelatihan terhadap setiap individu dan *offspring*.

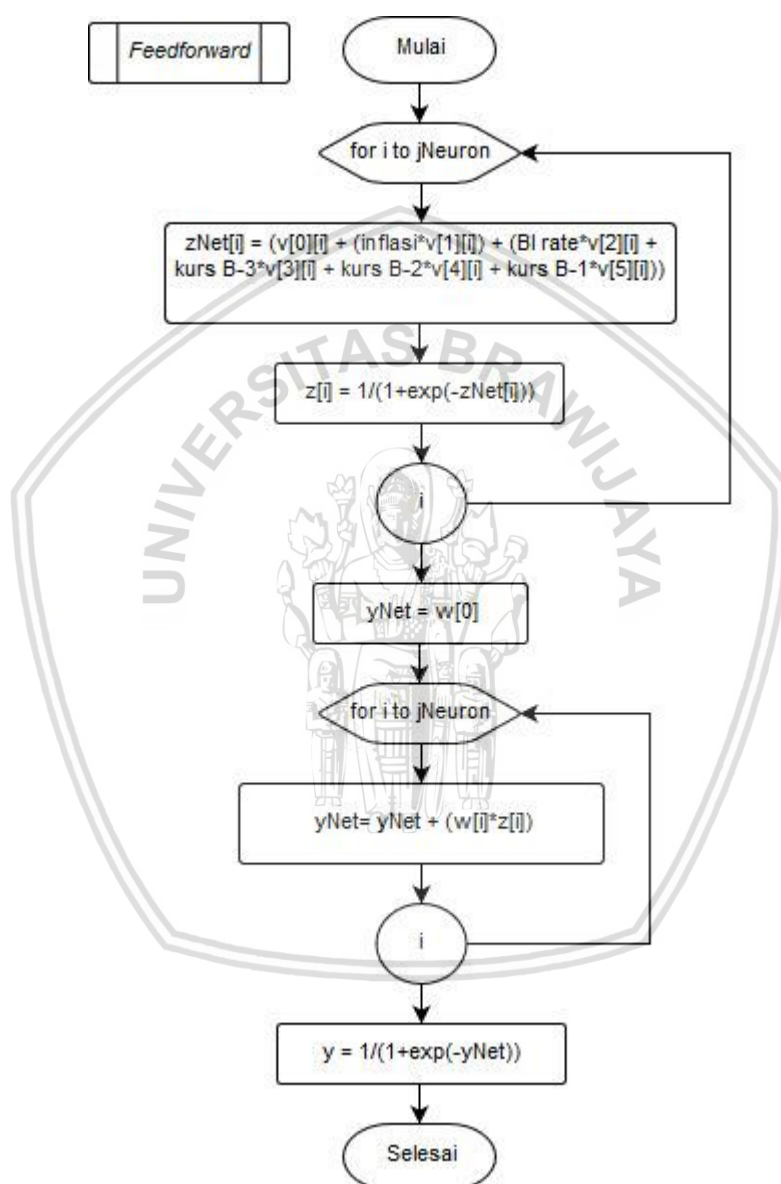
4.3.4 Pelatihan Individu dengan *Backpropagation*.

Pada tahap ini semua individu akan dilatih menggunakan *Backpropagation* dengan langkah-langkah seperti pada Gambar 4.15. Proses pelatihan diawali dengan melakukan inisialisasi array *v* yang akan menampung nilai bobot dari lapisan masukan terhadap lapisan tersembunyi, inisialisasi array *w* yang akan menampung nilai bobot dari lapisan tersembunyi terhadap lapisan keluaran. Inisialisasi array *z* dan *zNet* yang merupakan variable penampung nilai komputatif dari perkalian nilai masukan dan bobot dari lapisan masukan ke lapisan tersembunyi, inisialisasi *yNet* dan *y* sebagai variable yang akan menampung hasil jumlah dari perkalian antara hasil pada lapisan tersembunyi ke arah lapisan keluaran. δy merupakan variable yang akan menampung nilai error pada lapisan tersembunyi sedangkan δ dan δ_{Net} merupakan variable yang akan menampung nilai error pada lapisan tersembunyi. Δw dan Δv merupakan variable yang akan menyimpan besar perubahan pada bobot *w* dan *v*. Proses pelatihan pada *Backpropagation* dilakukan secara mundur, dimulai dari mencari error pada lapisan keluaran yang diikuti dengan penentuan perubahan besar bobot pada lapisan tersembunyi terhadap lapisan keluaran. Setelah itu mencari besar error pada lapisan tersembunyi dan kemudian mencari besar perubahan yang diperlukan. Hal ini terus dilakukan pada setiap data disetiap iterasi pada pelatihan *Backpropagation*.



Gambar 4.14 Diagram alur pelatihan *Backpropagation*

Bobot pada individu terpilih akan digunakan pada arsitektur *Backpropagation*, lalu dilakukan proses *feedforward*. Pada proses ini akan didapat nilai hasil prediksi berdasarkan bobot yang telah dimiliki oleh individu tersebut. Hasil ini kemudian yang akan dievaluasi yang mempengaruhi arah pembelajaran dari *Backpropagation* terhadap bobot individu terpilih tersebut. Secara lebih singkat, proses dalam *feedforward* dapat dilihat pada Gambar 4.15.



Gambar 4.15 Diagram alur *Feedforward*

Pada Gambar 4.15 dijelaskan alur proses *feedforward* yang awalnya setiap bobot dari masukan (v) ke lapisan tersembunyi dikalikan terhadap nilai masukan kemudian setiap hasil kali dijumlahkan yang memberikan hasil berupa $zNet$ seperti pada Persamaan 2.3. Hasil yang telah dijumlahkan kemudian dibangkitkan dengan fungsi aktivasi *sigmoid*. Contoh perhitungannya dilakukan pada *Parent 1* pada data kedua pada data latih adalah sebagai berikut

Parent 1

Parent	V10	V11	V12	V13	V14	V15	V20	V21	V22
Parent 1	0,7	0,2	0,5	0,7	0,1	0,83	0,87	0,36	0,84
V23	V24	V25	W10	W11	W12				
0,36	0,37	0,38	0,57	0,91	0,94				

Gambar 4.16 Kromosom dari Parent 1

Inflasi	BI rate	Perubahan Kurs 3 Bulan Sebelumnya	Perubahan Kurs 2 Bulan Sebelumnya	Perubahan Kurs 1 Bulan Sebelumnya	Perubahan Kurs Bulan ini (Target)
0,1	0,1	0,566206	0,580178	0,525444	0,17372

Gambar 4.17 Contoh data yang dihitung

$$Znet_1 = V_{10} + (V_{11} * inflasi) + (V_{12} * BI Rate) + (V_{13} * K3) + (V_{14} * K2) + (V_{15} * K1)$$

$$Znet_1 = 0,7 + (0,2 * 0,1) + (0,5 * 0,1) + (0,7 * 0,566206) + (0,1 * 0,580178) + (0,83 * 0,525444)$$

$$Znet_1 = 1,66048052$$

$$Znet_2 = V_{20} + (V_{21} * inflasi) + (V_{22} * BI Rate) + (V_{23} * K3) + (V_{24} * K2) + (V_{25} * K1)$$

$$Znet_2 = 0,87 + (0,36 * 0,1) + (0,84 * 0,1) + (0,36 * 0,566206) + (0,37 * 0,580178) + (0,38 * 0,525444)$$

$$Znet_2 = 1,608168817$$

$Znet$ yang sudah dihitung kemudian dibangkitkan dengan menggunakan fungsi aktivasi *sigmoid* untuk mendapatkan nilai Z_1 dan Z_2 .

$$Z_1 = \frac{1}{(1 + \exp(-Znet_1))}$$

$$Z_2 = \frac{1}{(1 + \exp(-Znet_2))}$$

$$Z_1 = \frac{1}{(1 + \exp(-1,660481))}$$

$$Z_2 = \frac{1}{(1 + \exp(-1,608168817))}$$

$$Z_1 = 0,873894$$

$$Z_2 = 0,85564957$$

Setelah mendapatkan nilai Z_1 dan Z_2 kemudian mencari nilai Y yang merupakan keluaran dari *Backpropagation*. Nilai Y didapatkan sesuai dengan Persamaan 2.5 dan Persamaan 2.6.

$$Ynet = W_{10} + (W_{11} * Z_1) + (W_{12} * Z_2)$$

$$Ynet = 0,57 + (0,91 * 0,680605) + (0,94 * 0,666314)$$

$$Y_{net} = 1,815686$$

$$Y = \frac{1}{(1 + \exp(-Y_{net}))}$$

$$Y = \frac{1}{(1 + \exp(-1,815686))}$$

$$Y = 0,720095$$

Setelah mendapatkan nilai keluaran dari metode *feedforward* pada *Backpropagation*, kemudian melakukan evaluasi dengan menghitung nilai *error* dari keluaran terhadap nilai target dan menghitung nilai *error* dari lapisan tersembunyi terhadap hasil keluaran. Nilai *error* yang telah didapat kemudian menjadi acuan terhadap besar perubahan yang perlu dilakukan terhadap bobot. Nilai δ didapatkan sesuai dengan Persamaan 2.7 dengan contoh penghitungan sebagai berikut.

$$\delta = (t - y) y(1 - y)$$

$$\delta = (0,17372 - 0,720095) * 0,720095 * (1 - 0,720095)$$

$$\delta = -0,102938272$$

Setelah mendapatkan *error* dari keluaran terhadap nilai target, kemudian menghitung besar perubahan bobot dari lapisan tersembunyi ke lapisan keluaran sesuai dengan Persamaan 2.8 dengan nilai alpha 0.1 maka contoh penghitungan dapat dituliskan sebagai berikut.

$$\Delta w_{kj} = \alpha \delta_k z_j$$

$$\Delta w_{10} = \alpha \delta_1 z_0$$

$$\Delta w_{10} = 0,1 * -0,102938272 * 1$$

$$\Delta w_{10} = -0,010293827$$

$$\Delta w_{11} = \alpha \delta_1 z_1$$

$$\Delta w_{12} = \alpha \delta_1 z_2$$

$$\Delta w_{11} = 0,1 * -0,102938272 * 0,873894$$

$$\Delta w_{12} = 0,1 * -0,102938272 * 0,85564957$$

$$\Delta w_{11} = -0,008995714$$

$$\Delta w_{12} = -0,008807909$$

Kemudian mencari nilai *error* dari lapisan masukan terhadap lapisan tersembunyi dengan melihat perubahan bobot dari lapisan tersembunyi ke lapisan keluaran sesuai dengan Persamaan 2.9 dan Persamaan 2.10.

$$\delta_{net_j} = \sum_{k=1}^m \delta_k \cdot w_{kj}$$

$$\delta_{net_1} = \delta \cdot w_{11}$$

$$\delta_{net_2} = \delta \cdot w_{12}$$

$$\delta_{net_1} = -0,102938272 * 0,2$$

$$\delta_{net_2} = -0,102938272 * 0,5$$

$$\delta_{net_1} = -0,02058765$$

$$\delta_{net_2} = -0,05147$$

$$\delta_j = \delta_{net_j} \cdot f'(z_{net_j}) = \delta_{net_j} \cdot z_j(1 - z_j)$$

$$\delta_1 = \delta_{net_1} \cdot z_1(1 - z_1)$$

$$\delta_1 = -0,02058765 * 0,873894 * (1 - 0,873894)$$

$$\delta_1 = -0,00227$$

$$\delta_2 = \delta_{net_2} \cdot z_2(1 - z_2)$$

$$\delta_2 = -0,05147 * 0,85564957 * (1 - 0,85564957)$$

$$\delta_2 = -0,00636$$

Setelah mendapatkan nilai *error* pada lapisan tersembunyi kemudian menghitung besar perubahan nilai bobot dari lapisan masukan terhadap lapisan tersembunyi dengan alpha 0.1 dan rumus sesuai dengan Persamaan 2.11.

$$\Delta v_{ji} = \alpha \delta_j x_i$$

$$\Delta v_{10} = 0,1 * (-0,00227) * 1$$

$$\Delta v_{10} = (-0,000227)$$

$$\Delta v_{20} = 0,1 * (-0,00636) * 1$$

$$\Delta v_{20} = (-0,000636)$$

$$\Delta v_{11} = 0,1 * (-0,00227) * 0,1$$

$$\Delta v_{11} = (-0,0000227)$$

$$\Delta v_{21} = 0,1 * (-0,00636) * 0,1$$

$$\Delta v_{21} = (-0,0000636)$$

Besar perubahan terus dihitung sebanyak jumlah bobot dari lapisan masukan ke lapisan tersembunyi. Besar perubahan yang telah didapat kemudian dijumlahkan dengan bobot sebelumnya sehingga mendapatkan bobot baru sesuai dengan Persamaan 2.12 dan Persamaan 2.13.

$$w_{kj}(\text{baru}) = w_{kj}(\text{lama}) + \Delta w_{kj}$$

$$w_{10}(\text{baru}) = w_{10}(\text{lama}) + \Delta w_{10}$$

$$w_{10}(\text{baru}) = 0,7 + (-0,010293827)$$

$$w_{10}(\text{baru}) = 0,689706$$

$$v_{ji}(\text{baru}) = v_{ji}(\text{lama}) + \Delta v_{ji}$$

$$v_{10}(\text{baru}) = v_{10}(\text{lama}) + \Delta v_{10}$$

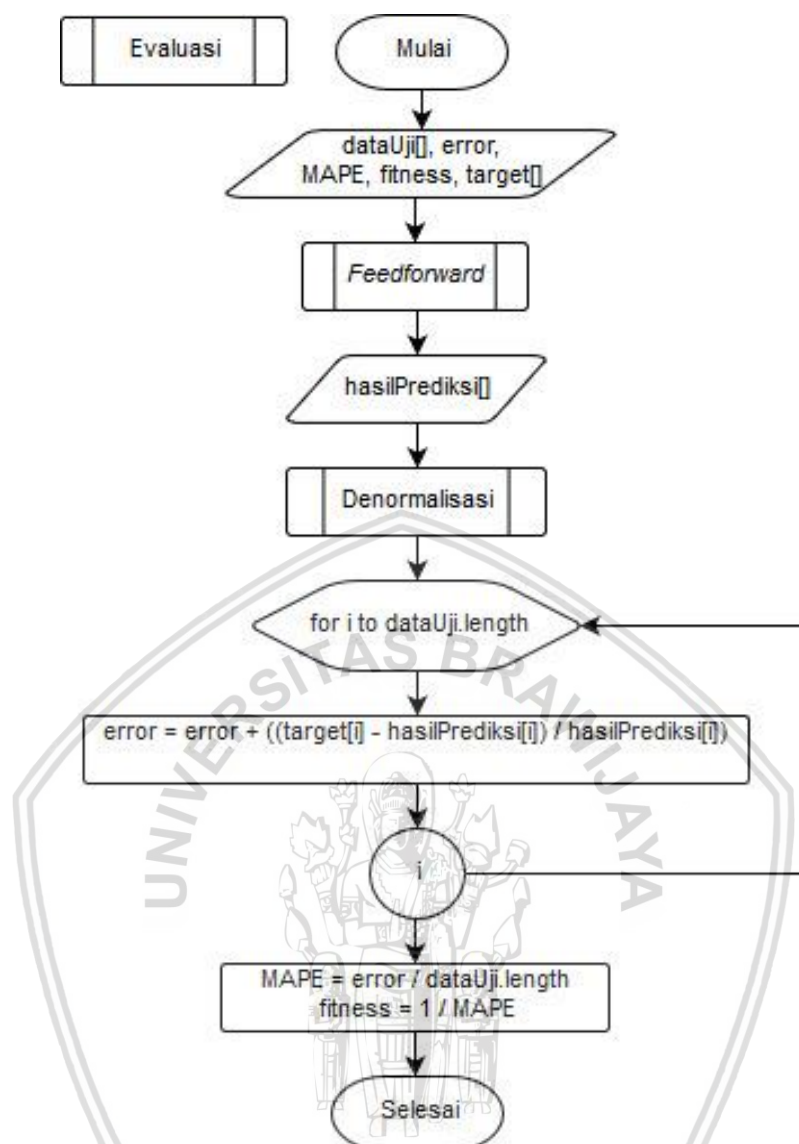
$$v_{10}(\text{baru}) = 0,7 + (-0,000227)$$

$$v_{10}(\text{baru}) = 0,699773$$

Perubahan bobot baru terus dilakukan hingga semua bobot selesai dijumlahkan dengan besar perubahan bobot yang telah didapatkan sebelumnya.

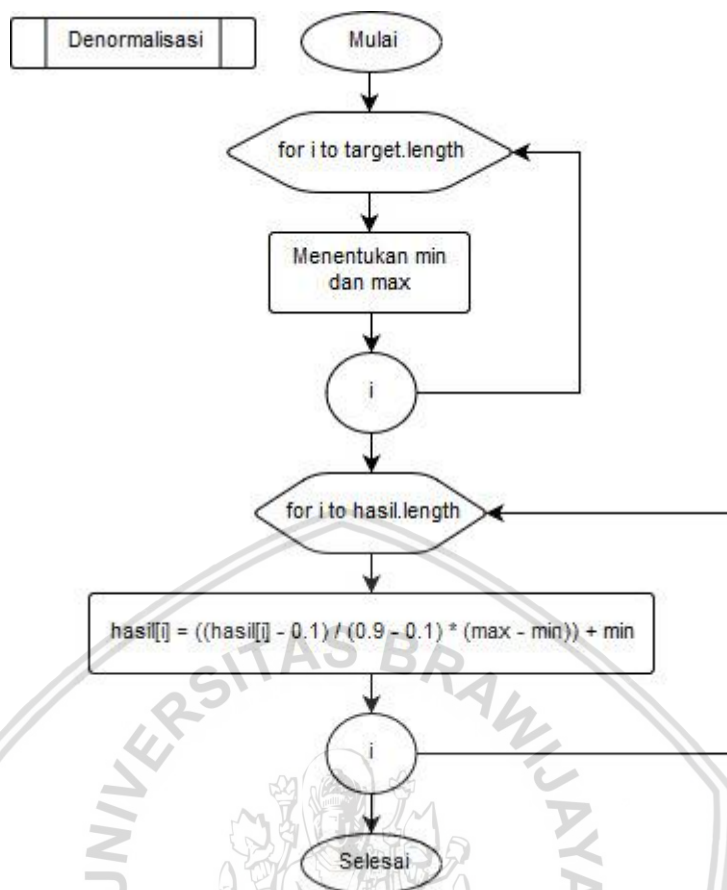
4.3.5 Evaluasi

Evaluasi merupakan proses untuk mendapatkan nilai *fitness* dari masing-masing individu. Proses evaluasi dilakukan dengan menghitung nilai MAPE dari hasil *Feedforward* pada metode *Backpropagation* dengan menggunakan bobot setiap individu. Bobot dari individu terpilih akan diujikan pada arsitektur *Backpropagation* dengan menggunakan data latih. Hasil prediksi dari setiap data latih tersebut kemudian dihitung nilai *error* untuk mendapatkan MAPE seperti pada Persamaan 2.20.



Gambar 4.18 Diagram alur evaluasi

Misalkan *Parent* 1 akan dilakukan penghitungan nilai *fitness*-nya penghitungan akan dilakukan pada setiap data dalam data latih sesuai Tabel 4.1 dengan bobot *Parent* 1 sesuai Tabel 4.3. Sesuai dengan Persamaan 2.3 hingga Persamaan 2.6 yang merupakan fase *feedforward* pada *Backpropagation* maka langkah penghitungan nilai *fitness* dapat dijabarkan seperti pada Gambar 4.19.



Gambar 4.19 Diagram alur denormalisasi

Misalkan hasil prediksi yang dilakukan pada proses *feedforward* yang telah dilakukan dengan menggunakan *Parent 1* menghasilkan data seperti pada Tabel 4.1, maka selanjutnya adalah melakukan denormalisasi pada hasil prediksi yang telah dilakukan pada proses *feedforward* yang secara ringkas dijelaskan pada Gambar 4.20.

Tabel 4.6 Hasil prediksi *Parent 1*

Inflasi	<i>BI rate</i>	Perubahan Kurs 3 Bulan Sebelumnya	Perubahan Kurs 2 Bulan Sebelumnya	Perubahan Kurs 1 Bulan Sebelumnya	Prediksi	Perubahan Kurs Bulan ini (Target)
0,0321	0,065	-13,2857	239,7929	-64,6045	0,294806	-236,222
0,0333	0,0675	-373,35	-322,557	-13,2857	0,287959	239,7929
0,0445	0,0675	182,0286	34,45	-373,35	0,298375	-322,557
0,0414	0,0725	-600,238	-123,286	182,0286	0,303395	34,45
0,0489	0,075	406,9605	614,3452	-600,238	0,317593	-123,286
0,0683	0,075	172,7118	61,55138	406,9605	0,330446	614,3452
0,0726	0,075	-119,056	192,7644	172,7118	0,324758	61,55138
0,0715	0,075	170,7469	316,9761	-119,056	0,324706	192,7644
0,0638	0,075	279,9857	140,8095	170,7469	0,328215	316,9761
0,0696	0,0775	254,0968	13,43043	279,9857	0,332548	140,8095

Hasil prediksi pada Tabel 4.6 kemudian dilakukan denormalisasi dengan nilai maksimum baru akan sama dengan nilai terbesar pada nilai target dan nilai minimum baru akan sama dengan nilai terkecil pada nilai target sehingga akan menghasilkan nilai seperti pada Tabel 4.7.

Tabel 4.7 Hasil denormalisasi prediksi *Parent 1*

Inflasi	<i>BI rate</i>	Perubahan Kurs 3 Bulan Sebelumnya	Perubahan Kurs 2 Bulan Sebelumnya	Perubahan Kurs 1 Bulan Sebelumnya	Prediksi (y)	Perubahan Kurs Bulan ini (T)
0,0321	0,065	-13,2857	239,7929	-64,6045	-94,4147	-236,222
0,0333	0,0675	-373,35	-322,557	-13,2857	-102,433	239,7929
0,0445	0,0675	182,0286	34,45	-373,35	-90,2348	-322,557
0,0414	0,0725	-600,238	-123,286	182,0286	-84,3558	34,45
0,0489	0,075	406,9605	614,3452	-600,238	-67,7273	-123,286
0,0683	0,075	172,7118	61,55138	406,9605	-52,6751	614,3452
0,0726	0,075	-119,056	192,7644	172,7118	-59,3367	61,55138
0,0715	0,075	170,7469	316,9761	-119,056	-59,3976	192,7644
0,0638	0,075	279,9857	140,8095	170,7469	-55,288	316,9761
0,0696	0,0775	254,0968	13,43043	279,9857	-50,2138	140,8095

Setelah dilakukan denormalisasi terhadap nilai prediksi, kemudian menghitung nilai MAPE sesuai dengan Persamaan 2.20. Dengan mencari total dari nilai selisih antara nilai prediksi dan target yang dibagi dengan nilai target, maka didapatkan hasil seperti pada Tabel 4.8.

Tabel 4.8 Hasil penghitungan MAPE dari *Parent 1*

Perubahan Kurs Bulan ini (T)	Prediksi (y)	Selisih (y - T)	$\frac{ y - T }{T}$
-236,222	-94,4147	141,8073	0,600314
239,7929	-102,433	342,2259	1,427174
-322,557	-90,2348	232,3222	0,720252
34,45	-84,3558	118,8058	3,448644
-123,286	-67,7273	55,5587	0,450649
614,3452	-52,6751	667,0203	1,085742
61,55138	-59,3367	120,8881	1,964019
192,7644	-59,3976	252,162	1,308136
316,9761	-55,288	372,2641	1,174423
140,8095	-50,2138	191,0233	1,356608
MAPE			1,353596

Pada Tabel 4.8 menunjukkan hasil penghitungan nilai MAPE dari *Parent 1*. MAPE tersebut kemudian digunakan untuk mendapatkan nilai *fitness* dari *Parent 1* sesuai dengan Persamaan 2.19.

$$fitness = \frac{1}{MAPE}$$

$$fitness = \frac{1}{1,353596}$$

$$fitness = 0,738773$$

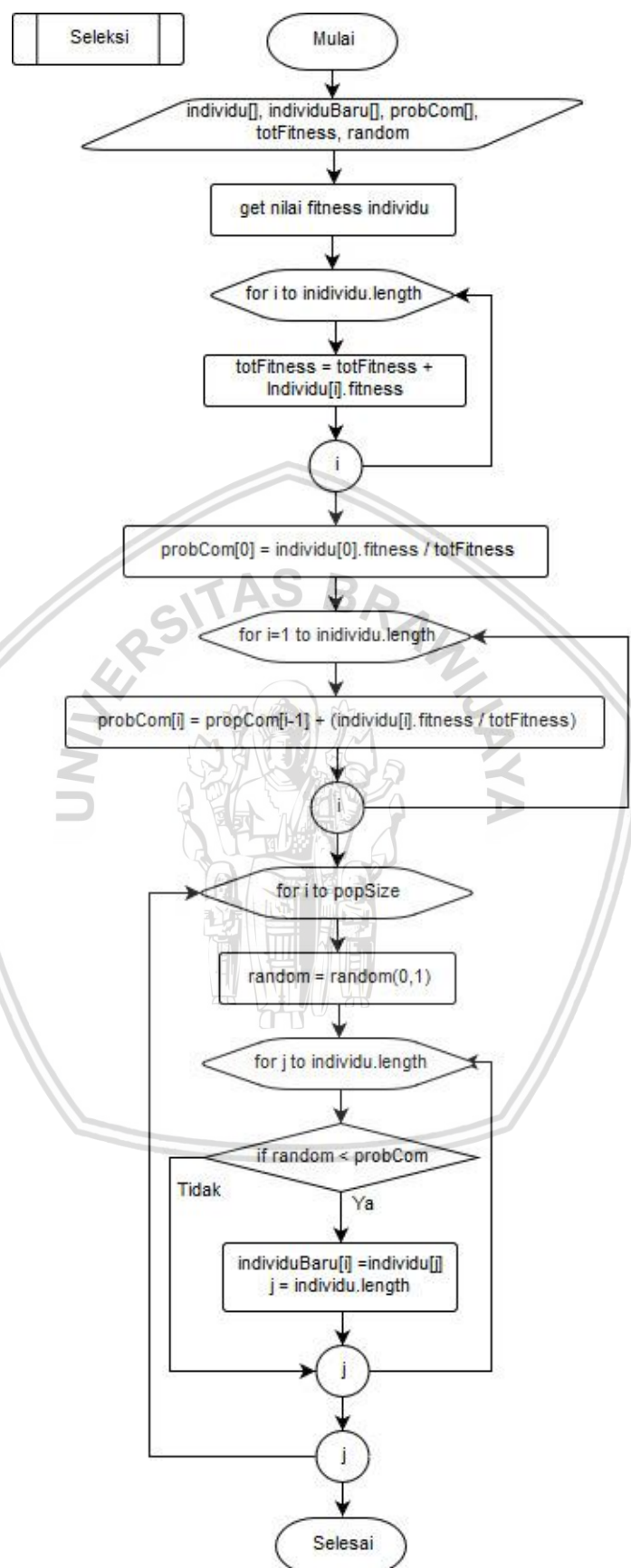
Penghitungan nilai *fitness* dilakukan terhadap seluruh individu dan *offspring* sehingga menghasilkan nilai seperti pada Tabel 4.9.

Tabel 4.9 *Fitness* semua individu

Individu	MAPE	<i>Fitness</i>
<i>Parent 1</i>	1,353596	0,738772853
<i>Parent 2</i>	1,734639	0,576488967
<i>Parent 3</i>	1,551431	0,644566152
<i>Parent 4</i>	1,147116	0,871751298
<i>Parent 5</i>	1,273456	0,785264806
<i>Parent 6</i>	1,071789	0,933019084
<i>Parent 7</i>	1,643085	0,608611296
<i>Parent 8</i>	1,125537	0,888464681
<i>Parent 9</i>	0,790308	1,265330145
<i>Parent 10</i>	0,771963	1,295399087
<i>Child 1</i>	1,145593	0,872910078
<i>Child 2</i>	0,788118	1,268844935
<i>Child 3</i>	1,117998	0,894456011

4.3.6 Seleksi

Seleksi akan dilakukan terhadap semua individu berdasarkan nilai *fitness*nya. Metode seleksi yang digunakan pada kasus ini adalah *stochastic sampling with replacement* atau yang lebih dikenal dengan *Roulette Wheel*. *Roulette Wheel* merupakan metode seleksi dengan menggunakan probabilitas kumulatif dari setiap individu dan *offspring*. Metode ini menggunakan nilai random untuk memilih individu atau *offspring* yang akan dipilih sebagai calon induk selanjutnya dengan melihat nilai probabilitas kumulatif tiap individu dan *offspring*. Berikut merupakan proses seleksi dengan menggunakan *Roulette Wheel*.



Gambar 4.20 Diagram alur seleksi *Roulette Wheel*

Nilai *fitness* setiap individu dan *offspring* yang telah didapatkan pada proses Evaluasi kemudian dihitung nilai totalnya dan setiap individu dan *offspring* dihitung probabilitasnya kemudian didapatkan nilai probabilitas kumulatifnya Misalkan telah didapat nilai *fitness* kumulatif seperti pada Tabel 4.10.

Tabel 4.10 Nilai *fitness* kumulatif

Individu	<i>Fitness</i>	Probabilitas	Probabilitas Komulatif
Parent 1	0,738772853	0,063447	0,063447
Parent 2	0,576488967	0,04951	0,112957
Parent 3	0,644566152	0,055357	0,168314
Parent 4	0,871751298	0,074868	0,243182
Parent 5	0,785264806	0,06744	0,310622
Parent 6	0,933019084	0,08013	0,390751
Parent 7	0,608611296	0,052269	0,44302
Parent 8	0,888464681	0,076303	0,519323
Parent 9	1,265330145	0,108669	0,627993
Parent 10	1,295399087	0,111252	0,739244
Child 1	0,872910078	0,074967	0,814211
Child 2	1,268844935	0,108971	0,923182
Child 3	0,894456011	0,076818	1
Total Fitness	11,64387939		

Setelah mendapatkan nilai *fitness* kumulatif dari setiap individu, sekarang dapatkan nilai *r* yang merupakan nilai random antara 0 hingga 1 sejumlah popSize yaitu seperti pada Tabel 4.11. Dengan hasil seleksi seperti pada Tabel 12.

Tabel 4.11 Nilai random

<i>R</i>	Nilai
r1	0,78
r2	0,56
r3	0,13
r4	0,65
r5	0,29
r6	0,95
r7	0,24

<i>R</i>	Nilai
r8	0,46
r9	0,41
r10	0,84

Tabel 4.12 Individu baru hasil seleksi

Individu	Individu asal	v01	v11	v21	v31	v41	v51	v02	v12	v22	v32	v42	v52	w01	w11	w21	fitness
Parent 1	Child 1	0,56	0,51	0,75	0,86	0,78	0,51	0,85	0,89	0,89	0,17	0,14	0,75	0,15	0,87	0,94	0,982364
Parent 2	Parent 9	0,74	0,94	0,99	0,65	0,46	0,75	0,19	0,14	0,75	0,74	0,56	0,18	0,85	0,95	0,46	0,633679
Parent 3	Parent 3	0,64	0,69	0,81	0,12	0,46	0,63	0,31	0,63	0,71	0,19	0,29	0,64	0,88	0,91	0,67	0,581763
Parent 4	Parent 10	0,65	0,15	0,31	0,47	0,31	0,86	0,97	0,88	0,56	0,47	0,49	0,85	0,43	0,95	0,73	0,836826
Parent 5	Parent 5	0,75	0,45	0,65	0,75	0,65	0,89	0,97	0,75	0,47	0,32	0,75	0,31	0,19	0,75	0,67	0,986847
Parent 6	Child 3	0,65	0,15	0,31	0,47	0,31	0,86	0,97	1,19	0,56	0,47	0,49	0,85	0,43	0,95	0,73	0,837101
Parent 7	Parent 4	0,54	0,13	0,64	0,41	0,71	0,64	0,65	0,46	0,19	0,76	18	0,82	0,84	0,65	0,64	0,663686
Parent 8	Parent 8	0,85	0,15	0,7	0,16	0,57	0,4	0,45	0,75	0,19	0,58	0,94	0,49	0,02	0,36	0,12	1,097784
Parent 9	Parent 6	0,75	0,38	0,48	0,29	0,59	0,933	0,83	0,65	0,64	0,36	0,59	0,65	0,55	0,91	0,95	0,773449
Parent 10	Child 2	0,6275	0,3732	0,5388	0,7469	0,5027	0,83655	0,9496	1,085	0,6788	0,278	0,3465	0,815	0,304	0,9428	0,7405	0,811296

4.4 Perancangan Antarmuka

Perancangan antarmuka akan menggambarkan antarmuka yang nantinya akan dirancang agar mudah digunakan dan dapat memenuhi kebutuhan sistem. Pada penelitian ini akan dibuat 3 antarmuka untuk mendukung jalannya sistem, yang meliputi proses optimasi pembelajaran sistem yang menggunakan *Backpropagation* dan Algoritme Genetika, antarmuka untuk menampilkan data yang akan digunakan untuk pembelajaran dan pengujian, serta antarmuka yang akan menampilkan hasil dari prediksi nilai tukar Rupiah terhadap dolar Amerika dengan menggunakan *Backpropagation* dan Algoritma Genetika.

4.4.1 Antarmuka Proses Optimasi Algoritme Genetika – *Backpropagation*

Antarmuka ini akan digunakan untuk menampilkan individu dari proses Algoritme Genetika dari setiap generasinya dengan menunjukkan setiap gen yang merupakan bobot bagi *Backpropagation* beserta nilai *fitness*-nya.

Pelatihan	Data	Hasil Uji
Title 1	Title 2	Title 3
		Title 4

Gambar 4.21 Antar Proses Optimasi *Backpropagation*-Algoritma Genetika

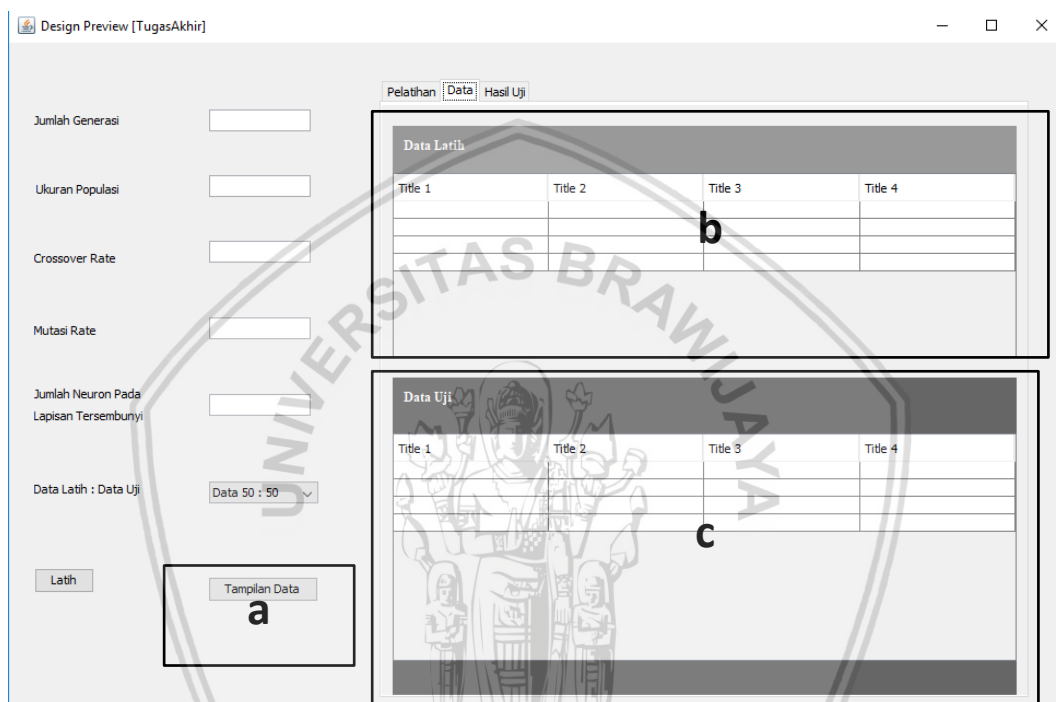
Keterangan dari antarmuka pada gambar 4.12 adalah sebagai berikut:

- Merupakan bagian untuk memenuhi kebutuhan sistem yang perlu diisi oleh user untuk mengisi parameter *Backpropagation* dan Algoritme Genetika. Parameter-parameter tersebut antara lain jumlah generasi, ukuran populasi, probabilitas *crossover*, probabilitas mutasi, dan jumlah *neuron* yang digunakan pada arsitektur *Backpropagation*.
- Combo box untuk menentukan perbandingan jumlah data yang akan ditampilkan atau digunakan sebagai data latih dan data uji.

- c. Tombol untuk melakukan proses optimasi *Backpropagation* dan Algoritme Genetika.
- d. Tabel yang akan menampilkan setiap individu dari setiap generasi beserta nilai *fitness*-nya.

4.4.2 Antarmuka Menampilkan Data

Antarmuka ini akan menampilkan data yang akan digunakan pada proses pelatihan dan pengujian. Antarmuka menampilkan data dapat dilihat pada gambar 4.12



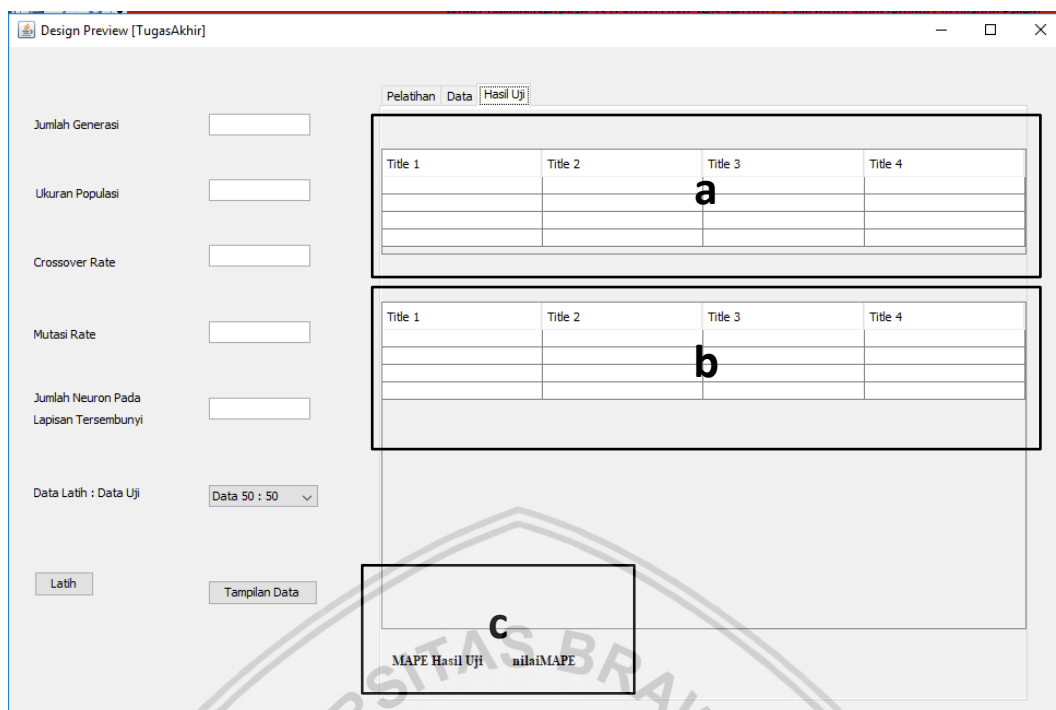
Gambar 4.22 Antarmuka Menampilkan Data

Keterangan dari antarmuka pada gambar 4.13 adalah sebagai berikut:

- a. Merupakan tombol untuk menampilkan isi data pada data latih dan data uji sesuai dengan perbandingan jumlah data yang telah dipilih pada combo box seperti pada kotak b pada gambar 4.12.
- b. Merupakan tabel yang akan menampilkan data latih.
- c. Merupakan tabel yang akan menampilkan data uji.

4.4.3 Antarmuka Menampilkan Hasil Pengujian

Antarmuka ini akan menampilkan hasil dari pengujian terhadap data uji dengan menggunakan individu terbaik hasil pelatihan yang telah dioptimasi dengan Algoritme Genetika. Pada antarmuka ini terdapat tabel yang akan menampilkan setiap data uji beserta hasil prediksi dan nilai sebenarnya. Untuk lebih jelaskan dapat dilihat pada gambar 4.14.



Gambar 4.23 Antarmuka Hasil Pengujian

Keterangan dari antarmuka pada gambar 4.14 adalah sebagai berikut:

- Merupakan tabel yang berisikan gen-gen dari individu terbaik hasil pelatihan *Backpropagation* yang telah dioptimasi dengan Algoritme Genetika
- Merupakan tabel yang akan menampilkan data uji beserta dengan nilai prediksi dan nilai sebenarnya
- Akan menampilkan hasil MAPE dari pengujian tersebut.

4.5 Perancangan Pengujian

Pengujian terhadap penerapan *Backpropagation* - Algoritme Genetika pada peramalan nilai tukar rupiah terhadap dolar Amerika merupakan tahapan untuk menemukan parameter terbaik untuk mendapatkan hasil prediksi dengan nilai *error* seminimal mungkin. Selain itu pengujian juga dimaksudkan untuk mengetahui besar pengaruh parameter yang diujikan terhadap hasil. Proses pengujian akan dilakukan dengan melakukan pelatihan bobot *Backpropagation* terlebih dahulu menggunakan Algoritme Genetika terhadap data latih dengan input parameter tertentu. Setelah pelatihan selesai dilakukan, bobot terbaik hasil Algoritme Genetika akan dievaluasi dengan menggunakan data uji, yang kemudian dihitung tingkat kesalahan pada peramalan.

Pengujian akan dilakukan dengan menggunakan sejumlah data latih nilai tukar rupiah terhadap dolar Amerika dengan melakukan perubahan pada beberapa parameter yaitu jumlah generasi, ukuran populasi, nilai *crossover rate* dan nilai *mutation rate*, dan jumlah neuron pada lapisan tersembunyi pada arsitektur *Backpropagation*.

4.5.1 Pengujian Jumlah Generasi

Pengujian jumlah generasi bertujuan untuk mendapatkan jumlah generasi yang dapat menghasilkan solusi paling optimal pada prediksi nilai tukar rupiah terhadap dolar Amerika. Pengujian akan dilakukan sebanyak 10 kali pada setiap percobaan. Jumlah generasi akan diujikan pada 10 generasi hingga 100 generasi. Pengujian jumlah generasi dijelaskan pada Tabel 4.9.

Tabel 4.13 Pengujian jumlah generasi

Jumlah Generasi	Percobaan ke-										Rataan <i>fitness</i>
	1	2	3	4	5	6	7	8	9	10	
10											
20											
30											
40											
50											
60											
70											
80											
90											
100											

4.5.2 Pengujian Ukuran Populasi

Pengujian ukuran populasi bertujuan untuk mendapatkan ukuran populasi yang dapat menghasilkan solusi paling optimal pada prediksi nilai tukar rupiah terhadap dolar Amerika. Pengujian akan dilakukan sebanyak 10 kali pada setiap percobaan. Ukuran populasi akan diujikan pada ukuran populasi 10 hingga ukuran populasi 100.

Tabel 4.14 Pengujian ukuran populasi

Ukuran Populasi	Percobaan ke-										Rataan fitness
	1	2	3	4	5	6	7	8	9	10	
10											
20											
30											
40											
50											
60											
70											
80											
90											
100											

4.5.3 Pengujian Nilai Crossover Rate dan Mutation Rate

Pengujian *crossover rate* dan *mutation rate* berujuan untuk mendapatkan nilai *crossover rate* dan *mutation rate* yang dapat menghasilkan solusi paling optimal pada prediksi nilai tukar rupiah terhadap dolar Amerika. Pengujian akan dilakukan sebanyak 10 kali pada setiap percobaan. Nilai yang akan diujikan akan ditunjukkan oleh Tabel 4.11.

Tabel 4.15 Pengujian nilai crossover rate dan mutation rate

Rate		Percobaan ke-										Rataan fitness
Crossover rate	Mutation Rate	1	2	3	4	5	6	7	8	9	10	
0,1	0,9											
0,2	0,8											
0,3	0,7											
0,4	0,6											
0,5	0,5											
0,6	0,4											
0,7	0,3											
0,8	0,2											
0,9	0,1											

4.5.4 Pengujian Jumlah *Neuron* pada Lapisan Tersembunyi *Backpropagation*

Pengujian jumlah lapisan tersembunyi pada *Backpropagation* berujuan untuk mendapatkan jumlah lapisan yang dapat menghasilkan solusi paling optimal pada prediksi nilai tukar rupiah terhadap dolar Amerika. Pengujian akan dilakukan sebanyak 10 kali pada setiap percobaan. Jumlah lapisan tersembunyi yang akan diujikan akan ditunjukan oleh Tabel 4.12.

**Tabel 4.16 Pengujian jumlah *neuron* pada lapisan tersembunyi
*Backpropagation***

Jumlah Neuron	Percobaan ke-										Rataan <i>fitness</i>
	1	2	3	4	5	6	7	8	9	10	
1											
2											
3											
4											
5											
6											
7											
8											
9											
10											
11											
12											
13											
14											
15											

BAB 5 PEMBAHASAN

Pada bab ini akan membahas tentang metode *Backpropagation* dan Algoritme Genetika yang telah diimplementasikan dengan menggunakan bahasa Java sesuai dengan perancangan yang telah dibuat sebelumnya. Pembahasan dalam bab ini memuat pembahasan mengenai lingkungan implementasi, deskripsi data, *source code* implementasi *Backpropagation*-Algoritme Genetika, serta implementasi antarmuka.

5.1 Lingkungan Implementasi

Lingkungan implementasi pada penelitian ini terbagi atas lingkungan perangkat lunak, dan lingkungan perangkat keras.

5.1.1 Lingkungan Perangkat Keras

Untuk implementasi metode *Backpropagation* dan *Genetic Algorithm* digunakan perangkat keras dengan spesifikasi yang digunakan sebagai berikut:

Processor	AMD Quad-Core Processor A8-4500M up to 2.8 GHz
Memory	4GB DDR3 Memory
Harddisk	500 GB HDD
Monitor	14 inch

5.1.2 Lingkungan Perangkat Lunak

Untuk implementasi metode *Backpropagation* dan *Genetic Algorithm* digunakan perangkat lunak dengan spesifikasi yang digunakan sebagai berikut:

Sistem Operasi	Windows 10
Code Editor	Netbeans 8.2
IDE	Netbeans 8.2

5.2 Deskripsi Data

Data yang digunakan merupakan data Inflasi, *BI rate* dan nilai kurs bulanan dari halaman resmi Bank Indonesia yang terdiri dari 124 data dengan data latih sebanyak 99 data dan data latih sebanyak 25 data.

5.3 Kode Program Implementasi Algoritme Genetika – *Backpropagation*

Pada sub bab ini akan membahas implementasi dari metode Algoritme Genetika – *Backpropagation* yang terdiri dari inisialisasi populasi, reproduksi *crossover*, dan reproduksi mutasi, pelatihan setiap individu dengan *Backpropagation*, evaluasi dan terakhir seleksi dengan menggunakan *Roulette Wheel*.

5.3.1 Fungsi Inisialisasi Populasi

Pada fungsi inisialisasi populasi setiap gen pada setiap individu yang terbentuk akan diberikan bobot acak. Bobot yang terbentuk bergantung pada jumlah neuron pada lapisan tersembunyi yang ditentukan. Pemberian nilai bobot pada setiap gen dilakukan dengan memberikan nilai acak antara -0.1 hingga 0.1. Kode Program 5.1 merupakan implementasi fungsi inisialisasi populasi.

```

1 public void inisialisasi() {
2     for (int i = 0; i < popSize; i++) {
3         for (int j = 0; j < jGen - 1; j++) {
4             individu[i][j] = Math.random() * 2 - 1;
5         }
6         namaIndividu[i][0] = "P" + (i + 1);
7         namaIndividu[i][1] = "Individu" + (countIndividu++);
8         countIndividu++;
9     }
10 }

```

Kode Program 5.1 Fungsi Inisialisasi Populasi

5.3.2 Fungsi Reproduksi Crossover

Pada fungsi reproduksi *crossover* akan ditentukan dua individu yang dipilih secara acak untuk diproses sebagai induk sehingga nantinya akan menghasilkan *offspring*. Setiap proses akan menghasilkan dua *offspring*. Proses pemilihan individu sebagai induk dan pembentukan *offspring* baru akan terus dilakukan hingga jumlah yang telah ditentukan yang bergantung pada nilai *popSize* dan *CRate*. Kode Program 5.2 merupakan implementasi fungsi reproduksi *crossover* untuk menentukan individu yang akan dipilih sebagai induk, dan Kode Program 5.3 merupakan implementasi fungsi reproduksi *crossover* untuk mendapatkan nilai bobot bagi *offspring* baru yang terbentuk.

```

1 public void prosesCrossover(int iterasi) {
2     int parent1 = 0;
3     int parent2 = 0;
4     double[] a = new double[jGen];
5     for (int i = 0; i < iterasi; i++) {
6         namaIndividu[i + popSize][0] = "C" + (i + 1);
7         if (i % 2 == 0) {
8             parent1 = nilaiRandom(0, popSize);
9             parent2 = nilaiRandom(0, popSize);
10            for (int j = 0; j < jGen; j++) {
11                a[j] = Math.random();
12            }
13            prosesHitungCrossover(individu[parent1],
14            individu[parent2], a);
15        } else {
16            prosesHitungCrossover(individu[parent2],
17            individu[parent1], a);
18        }
19    }
20 }

```

Kode Program 5.2 Fungsi Reproduksi Crossover Pemilihan Induk


```

1 public void prosesHitungCrossover(double[] parent1, double[]
  parent2, double[] a) {
2     for (int j = 0; j < jGen - 1; j++) {
3         namaIndividu[countIndividu1][0] = "Cc" +
        (countIndividu1 - popSize);
4         individu[countIndividu1][j] = (parent1[j] + (a[j] *
        (parent2[j] - parent1[j])));
5     }
6 }

```

Kode Program 5.3 Fungsi Reproduksi Crossover Penghitungan Bobot Offspring

5.3.3 Fungsi Reproduksi Mutasi

Pada fungsi reproduksi mutasi akan ditentukan satu individu yang dipilih secara acak yang akan digunakan sebagai induk bagi *offspring* baru yang akan terbentuk. Setiap proses akan menghasilkan satu *offspring* dari satu induk. Proses pemilihan individu sebagai induk dan pembentukan offspring akan terus dilakukan hingga jumlah yang telah ditentukan yang bergantung pada nilai *popSize* dan *MRate*. Kode Program 5.4 merupakan implementasi fungsi reproduksi mutasi untuk menentukan individu yang akan dipilih sebagai induk, dan Kode Program 5.5 merupakan implementasi fungsi reproduksi mutasi untuk mendapatkan nilai bobot bagi *offspring* baru yang terbentuk.

```

1 public void prosesMutasi(int iterasi) {
2     int parent, cm;
3     int index = 0;
4     for (int i = 0; i < iterasi; i++) {
5         parent = nilaiRandom(0, popSize);
6         cm = nilaiRandom(0, 20);
7         prosesHitungMutasi(individu[parent], cm);
8     }
9 }
10 }

```

Kode Program 5.4 Fungsi Reproduksi Mutasi Pemilihan Induk

```

1 public void prosesHitungMutasi(double[] parent, int cm) {
2     double r = Math.random();
3     for (int i = 0; i < jGen - 1; i++) {
4         if (i == cm) {
5             individu[countIndividu1][i] = parent[i] + r;
6         } else {
7             individu[countIndividu1][i] = parent[i];
8         }
9         namaIndividu[countIndividu1][0] = "Cm" +
        (countIndividu1 + popSize + Math.ceil(popSize * cRate));
10    }
11    namaIndividu[countIndividu1][1] = "Individu" +
        countIndividu++;
12    countIndividu1++;
13 }

```

Kode Program 5.5 Fungsi Reproduksi Mutasi Penghitungan Bobot Offspring

5.3.4 Fungsi Pelatihan Individu dengan *Backpropagation*

Fungsi pelatihan individu dengan *Backpropagation* dilakukan untuk mendapatkan nilai *fitness* yang optimum. Setiap individu akan dilatih dengan menerapkan alur maju (*feedforward*) dan alur mundur (*Backpropagation*) pada algoritma *Backpropagation*. Kode Program 5.6 merupakan implementasi fungsi pelatihan individu dengan *Backpropagation*.

```

1 public double[] latih(double[] individu) {
2     double alpha = this.alpha;
3     int iterasi = this.jIterasi;
4     int hitung = 0;
5     double fitness = 0;
6     double error;
7     double terbaik = hitungFitnessLatih(individu);
8     double sebelum = hitungFitnessLatih(individu);
9     double[] z = new double[jNeuron];
10    double y;
11    double[] deltaW = new double[jNeuron + 2];
12    double[] deltaV = new double[((jInput + 1) * jNeuron) +
    (jNeuron)];
13    double[] temp = new double[individu.length];
14    int l = 0;
15    for (int i = 0; i < individu.length; i++) {
16        temp[i] = individu[i];
17    }
18    for (l = 0; l < iterasi; l++) {
19        error = 0;
20        for (int i = 0; i < dataLatih.length; i++) {
21            y = 0;
22            for (int j = 0; j < jNeuron; j++) {
23                z[j] = 0;
24                for (int k = 0; k <= jInput; k++) {
25                    if (k == 0) {
26                        z[j] = z[j] + (temp[(jInput + 1) * j + k]
27 * 1);
28                    } else {
29                        z[j] = z[j] + (temp[(jInput + 1) * j + k]
30 * dataLatih[i][k - 1]);
31                    }
32                }
33            }
34            y = y + temp[(jInput + 1) * jNeuron];
35            for (int j = 0; j < jNeuron; j++) {
36                z[j] = 1 / (1 + Math.exp(-1 * z[j]));
37                y = y + (z[j] * temp[(jInput + 1) * jNeuron +
38 (j + 1)]);
39            }
40            y = 1 / (1 + (Math.exp(-1 * y)));
41            error = error + Math.pow((dataLatih[i][jInput] -
42 y), 2);
43            deltaW[deltaW.length - 1] = (dataLatih[i][jInput]
44 - y) * y * (1 - y);
45            deltaW[0] = alpha * deltaW[deltaW.length - 1] * 1;
46            for (int j = 0; j < deltaW.length - 2; j++) {
47                deltaW[j + 1] = alpha * deltaW[deltaW.length -
48 1] * z[j];
49            }
50            deltaV[deltaV.length - jNeuron + j] =

```

```

44      deltaW[deltaW.length - 1] * temp[(jInput + 1) * jNeuron + (j
      + 1)];
      deltaV[deltaV.length - jNeuron + j] =
      deltaV[deltaV.length - jNeuron + j] * z[j] * (1 - z[j]);
45      }
46      for (int j = 0; j < jNeuron; j++) {
47          for (int k = 0; k < (jInput + 1); k++) {
48              if (k == 0) {
49                  deltaV[j * (jInput + 1)] = alpha
* deltaV[deltaV.length - jNeuron + j] * 1;
50              } else {
51                  deltaV[(j * (jInput + 1)) + k] =
      alpha * deltaV[deltaV.length - jNeuron + j] * dataLatih[i][k
      - 1];
52              }
53          }
54      }
55      int count = 0;
56      for (int j = 0; j < deltaV.length - jNeuron;
      j++) {
57          count++;
58          temp[j] = temp[j] + deltaV[j];
59      }
60      for (int j = 0; j < deltaW.length - 1; j++) {
61          temp[count + j] = temp[count + j] +
      deltaW[j];
62      }
63      }
64      fitness = hitungFitnessLatih(temp);
65      error = 0.5 * error;
66      if (error < 0.1 || Math.abs(sebelum - fitness) <
      0.00001) {
67          individu = temp;
68          break;
69      }
70      sebelum = fitness;
71  }
72  return individu;
73  }

```

**Kode Program 5.6 Implementasi Fungsi Pelatihan Individu Dengan
Backpropagation**

5.3.5 Fungsi Evaluasi

Pada fungsi evaluasi akan dihitung nilai *fitness* dari setiap individu. Nilai *fitness* didapat dari hasil MAPE pada *feedforward* pada metode *Backpropagation* dari setiap data yang ada. Kode Program 5.7 merupakan implementasi dari fungsi evaluasi.

```

1  public double hitungFitnessLatih(double[] individu) {
2      double fitness, mape, mse;
3      fitness = 0;
4      mape = 0;
5      double[] z = new double[jNeuron];
6      double hasil = 0;
7      double error = 0;
8      double error1 = 0;
9      int count = 0;

```

```

10 double[] y = new double[dataLatih.length];
11 double jumlahTarget = 0;
12 for (int i = 0; i < dataLatih.length; i++) {
13     y[i] = 0;
14     for (int j = 0; j < jNeuron; j++) {
15         z[j] = 0;
16         for (int k = 0; k <= jInput; k++) {
17             if (k == 0) {
18                 z[j] = z[j] + (individu[((jInput + 1) *
19 j) + k] * 1);
20             } else {
21                 z[j] = z[j] + (individu[((jInput + 1) *
22 j) + k] * dataLatih[i][k - 1]);
23             }
24         }
25         y[i] = y[i] + individu[(jInput + 1) * jNeuron];
26         for (int j = 0; j < jNeuron; j++) {
27             z[j] = 1 / (1 + Math.exp(-1 * z[j]));
28             y[i] = y[i] + (z[j] * individu[(jInput + 1) *
29 jNeuron + (j + 1)]);
30         }
31         hasil = 1 / (1 + Math.exp(-1 * y[i]));
32         error = error +
33 ((Math.abs((denormalisasi(rawDataLatih, hasil, jInput) +
34 rawDataLatih[i][jInput + 1]) - (rawDataLatih[i][jInput] +
35 rawDataLatih[i][jInput + 1])) / (denormalisasi(rawDataLatih,
36 hasil, jInput) + rawDataLatih[i][jInput + 1])));
37         count++;
38     }
39     mape = error * 100 / count;
40     fitness = 1 / mape;
41     if (fitness > terbaik[jGen - 1]) {
42         terbaik = individu;
43     }
44     return fitness;
45 }

```

Kode Program 5.7 Implementasi Fungsi Evaluasi

5.3.6 Fungsi Seleksi

Pada fungsi seleksi seluruh individu yang terbentuk akan diseleksi sedemikian rupa sehingga diharapkan individu terbaiklah atau individu yang mampu memberikan *offspring* lebih baiklah yang akan terpilih untuk diproses pada generasi selanjutnya. Fungsi seleksi yang digunakan merupakan fungsi seleksi *Roulette Wheel* dimana setiap individu akan dicari nilai probabilitasnya terhadap seluruh individu, dan memilih individu yang akan lolos ke generasi selanjutnya dengan memilih probabilitas kumulatifnya. Kode Program 5.8 merupakan implementasi dari fungsi seleksi.

```

1 public void seleksi() {
2     double[][] temp = new double[individu.length][jGen];
3     String[][] namaTemp = new
String[namaIndividu.length][2];

```

```

4      for (int i = 0; i < temp.length; i++) {
5          for (int j = 0; j < temp[i].length; j++) {
6              temp[i][j] = 0;
7          }
8          namaTemp[i][0] = "individu";
9          namaTemp[i][1] = "individu";
10     }
11     boolean sama = true;
12     double jumlah = 0;
13     double random;
14     for (int i = 0; i < popSize; i++) {
15         double[] probKomulatif = new
double[individu.length];
16         for (int j = 0; j < individu.length; j++) {
17             jumlah = jumlah + individu[j][jGen - 1];
18         }
19         probKomulatif[0] = (individu[0][jGen - 1] /
jumlah);
20
21         for (int j = 1; j < probKomulatif.length; j++) {
22             probKomulatif[j] = probKomulatif[j - 1] +
(individu[j][jGen - 1] / jumlah);
23         }
24         sama = true;
25         while (sama) {
26             random = Math.random();
27             int index = 0;
28             for (int j = 0; j < individu.length; j++) {
29                 if (random <= probKomulatif[j]) {
30                     index = j;
31                     j = 999999999;
32                 }
33             }
34             if (!cek(namaTemp, namaIndividu[index][1])) {
35                 temp[i] = individu[index];
36                 namaTemp[i][0] = "P " + (i + 1);
37                 namaTemp[i][1] = namaIndividu[index][1];
38                 individu[index][jGen - 1] = 0;
39                 sama = false;
40             }
41         }
42     }
43     countIndividu1 = popSize;
44     individu = temp;
45     namaIndividu = namaTemp;
46 }

```

Kode Program 5.8 Implementasi Fungsi Seleksi

5.4 Implementasi Antarmuka

5.4.1 Tab Data

Tab data berisikan dua tabel yang menampilkan data latih dan data uji. Lebih jelas dapat dilihat pada Gambar 5.1

Prediksi Nilai Tukar Rupiah Terhadap Dolar Amerika
Dengan Backpropagation - Algoritma Genetika

Data Backpropagation Backpropagation - Algoritma Genetika Hasil

Data Latih

No.	Inflasi	Bi Rate	Beda Kurs 3 Bulan Sebelum...	Beda Kurs 2 Bulan Sebelum...	Beda Kurs 1 Bulan Sebelum...	Target
1	0.0321	0.065	-13.28571429	239.7928571	-64.60454545	-236.2219251
2	0.0333	0.0675	-373.35	-322.5571429	-13.28571429	239.7928571
3	0.0445	0.0675	182.0285714	34.45	-373.35	-322.5571429
4	0.0414	0.0725	-600.2380952	-123.2857143	182.0285714	34.45
5	0.0489	0.075	406.9605263	614.3452381	-600.2380952	-123.2857143
6	0.0683	0.075	172.7117794	614.3452381	406.9605263	614.3452381
7	0.0726	0.075	-119.0562771	192.764411	172.7117794	614.3452381
8	0.0715	0.075	170.7458972	316.8760766	-119.0562771	192.764411
9	0.0638	0.075	279.9857143	140.8095238	170.7458972	316.8760766

Data Uji

No.	Inflasi	Bi Rate	Beda Kurs 3 Bulan Sebelum...	Beda Kurs 2 Bulan Sebelum...	Beda Kurs 1 Bulan Sebelum...	Target
1	0.0345	0.065	-322.5571429	-13.28571429	239.7928571	-64.60454545
2	0.036	0.0675	34.45	-373.35	-322.5571429	-13.28571429
3	0.0718	0.075	192.764411	172.7117794	614.3452381	406.9605263
4	0.0836	0.0775	184.1060806	254.0963379	13.43043479	279.9857143
5	0.1106	0.0925	191.4	707.7	1662.8	-386.3078947
6	0.0725	0.075	82.55	-244.55	-508.05	8.7
7	0.0904	0.0725	685.5394737	-3.2	-824.45	-632.45
8	0.0275	0.065	-632.45	-186.0136364	-95.3030303	-133.7333333
9	0.0879	0.0875	36.86363636	120.6172249	191.8649886	499.1086957
10	0.043	0.0575	65.50789474	30.78636364	30.81363636	17.93688889

Tampilkan

Gambar 5.1 Tab Data

5.4.2 Tab Pelatihan Backpropagation

Prediksi Nilai Tukar Rupiah Terhadap Dolar Amerika
Dengan Backpropagation - Algoritma Genetika

Data Backpropagation Backpropagation - Algoritma Genetika Hasil

Backpropagation

Jumlah Neuron Pada Lapisan Tersembunyi: 2

Alpha Rate: 0.1

Jumlah Iterasi: 100

Proses

No.	Inflasi	Bi Rate	Beda Kur.	Beda Kur.	Beda Kur.	Target	Z_net1	Z_net2	Z1	Z2	Y_net	Y
57	0.467523	0.833333	0.392028	0.461522	0.427505	0.299934	-0.152648	-0.191217	0.461911	0.452340	-0.486984	0.364
58	0.473278	0.833333	0.409986	0.392028	0.630225	0.346913	-0.152789	-0.139230	0.461876	0.465248	-0.490219	0.37
59	0.573586	0.58	0.387474	0.388575	0.331690	0.366396	-0.138805	-0.202570	0.465851	0.449529	-0.491078	0.37
60	0.510277	0.58	0.415742	0.387474	0.501198	0.292742	-0.136232	-0.167314	0.465994	0.458268	-0.491906	0.37
61	0.509989	0.58	0.299234	0.415742	0.499250	0.369575	-0.134245	-0.181272	0.466489	0.459789	-0.494937	0.37
62	0.869578	0.9	0.322636	0.360607	0.568699	0.592801	-0.233749	-0.052834	0.441827	0.479303	-0.452852	0.37
63	0.808735	0.686666	0.366396	0.372796	0.506553	0.366757	-0.165750	-0.103811	0.458656	0.474070	-0.487210	0.38
64	0.201952	0.366666	0.364147	0.1	0.1	0.305346	-0.088115	-0.384191	0.477985	0.405116	-0.486846	0.38
65	0.502877	0.473333	0.344475	0.555673	0.457889	0.1	-0.083442	-0.146223	0.476556	0.453501	-0.491616	0.37
66	0.608941	0.833333	0.9	0.240924	0.370130	0.585673	-0.162494	-0.221246	0.459465	0.444512	-0.498430	0.37
67	0.215107	0.26	0.361066	0.361252	0.356096	0.388579	-0.039269	-0.263137	0.490183	0.434592	-0.494336	0.37
68	0.130421	0.26	0.340449	0.230732	0.452540	0.361252	-0.044741	-0.270822	0.488816	0.432705	-0.493786	0.37
69	0.113155	0.26	0.334523	0.322162	0.416072	0.230732	-0.041897	-0.273019	0.489527	0.432165	-0.494433	0.37
70	0.431346	0.26	0.415312	0.353988	0.403538	0.340187	-0.034903	-0.202105	0.491274	0.449645	-0.500394	0.37
71	0.317060	0.26	0.309055	0.318090	0.548490	0.353988	-0.035982	-0.186537	0.491005	0.453500	-0.501771	0.37
72	0.357348	0.313333	0.315683	0.329738	0.405478	0.367813	-0.054067	-0.221580	0.486486	0.444830	-0.501683	0.37
73	0.408324	0.313333	0.369924	0.325027	0.373859	0.329738	-0.054007	-0.221633	0.485501	0.444817	-0.501999	0.37
74	0.464234	0.313333	0.368545	0.392270	0.468207	0.325027	-0.049029	-0.171393	0.487744	0.457256	-0.504294	0.37
75	0.194552	0.1	0.403755	0.389750	0.471567	0.338357	-0.015476	-0.222190	0.503869	0.444679	-0.507219	0.37
76	0.213463	0.153333	0.440279	0.406905	0.528047	0.388750	-5.210754	-0.202404	0.499869	0.449570	-0.508286	0.37
77	0.265262	0.26	0.352582	0.364778	0.592651	0.406905	-0.034826	-0.180108	0.491294	0.455094	-0.507158	0.37
78	0.280883	0.1	0.366828	0.379087	0.497646	0.375078	0.017416	-0.194879	0.504354	0.451433	-0.507355	0.37
79	0.278417	0.1	0.402080	0.416929	0.462908	0.379087	0.018161	-0.201727	0.504540	0.449738	-0.507322	0.37
80	0.274306	0.1	0.410068	0.368446	0.525085	0.416929	0.016883	-0.190919	0.504220	0.452414	-0.507299	0.37
81	0.386947	0.153333	0.372500	0.369880	0.480781	0.403971	0.001334	-0.179966	0.500333	0.455129	-0.505595	0.37
82	0.359815	0.1	0.378506	0.364956	0.472764	0.369880	0.017979	-0.186596	0.504494	0.453485	-0.505021	0.37
83	0.226477	0.1	0.375087	0.372048	0.493387	0.364956	0.037947	-0.187329	0.504486	0.453293	-0.505021	0.37

Gambar 5.2 Tab Pelatihan Backpropagation

Pada tab pelatihan *Backpropagation* akan ditampilkan kolom untuk mengisi parameter yang diperlukan untuk melakukan pelatihan menggunakan *Backpropagation* yang diantaranya adalah “Jumlah Neuron Pada Lapisan Tersembunyi”, “Alpha Rate”, dan “Jumlah Iterasi”. Terdapat tombol proses untuk memberikan perintah untuk menjalankan proses pelatihan dengan parameter yang

telah dimasukkan, selain itu juga terdapat tabel di sisi kanan yang akan menunjukkan hasil dari setiap pelatihan terhadap setiap data di setiap iterasi. Gambar 5.2 menunjukkan tab pelatihan *Backpropagation* dengan lebih detail.

5.4.3 Tab Pelatihan Algoritme Genetika – *Backpropagation*

Pada tab pelatihan Algoritme Genetika – *Backpropagation* ditampilkan beberapa kolom masukan yang nantinya akan digunakan sebagai nilai parameter pada pelatihan menggunakan metode Algoritme Genetika – *Backpropagation* diantaranya adalah “Jumlah Generasi”, “Ukuran Populasi”, “Crossover Rate”, “Mutation Rate”, “Jumlah Neuron Pada Lapisan Tersembunyi”, “Alpha Rate”, dan “Jumlah Iterasi”. Lebih jelas dapat dilihat pada Gambar 5.3.

Prediksi Nilai Tukar Rupiah Terhadap Dolar Amerika
Dengan Backpropagation - Algoritma Genetika

Tab: Data | Backpropagation | Backpropagation - Algoritma Genetika | Hasil

Backpropagation - Algoritma Genetika

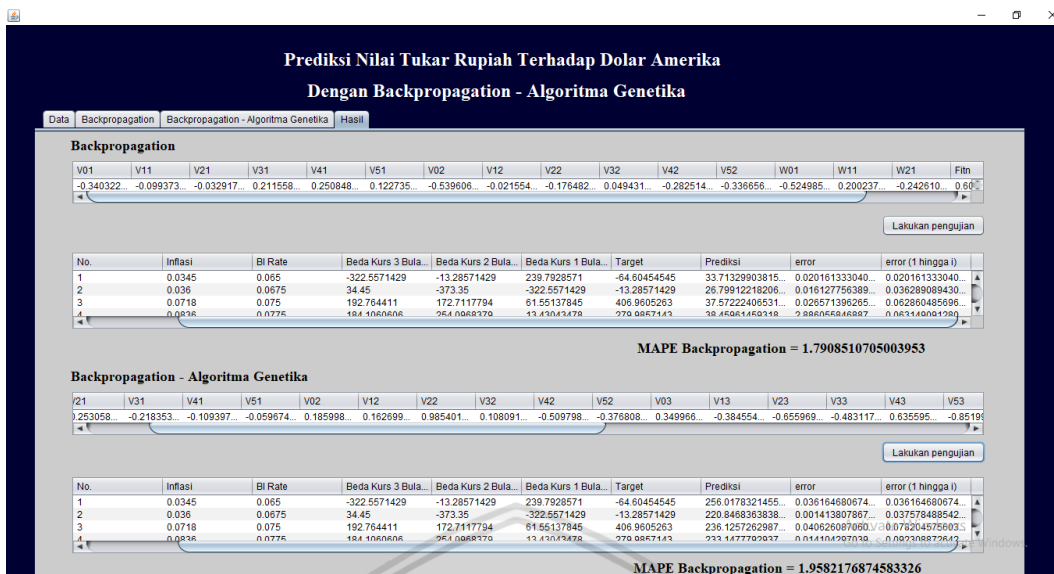
Jumlah Generasi: 10
 Ukuran Populasi: 10
 Crossover Rate: 0.6
 Mutasi Rate: 0.001
 Jumlah Neuron Pada Lapisan Tersembunyi: 4
 Alpha Rate: 0.1
 Jumlah Iterasi: 1000

No.	Nama	V01	V11	V21	V31	V41	V51	V02	V12	V22	V32
GENERASI 1											
1	P1	0.918740...	0.893782...	0.611319...	0.971538...	0.004378...	0.068730...	1.078400...	-0.905592...	-0.694988...	-0.8388...
2	P2	-0.514246...	0.431651...	0.512612...	-0.778990...	0.159256...	0.642723...	0.901508...	-0.296313...	0.484905...	-0.2806...
3	P3	0.798856...	-0.283778...	0.698747...	-0.848541...	-0.862651...	-0.236591...	-0.154381...	-0.238478...	1.032444...	-0.4465...
4	P4	-0.480324...	-0.501221...	0.879387...	-0.059122...	-0.442208...	-0.240112...	-0.917649...	-0.400705...	-0.125951...	0.8066...
5	P5	-0.495522...	-0.874154...	0.930123...	0.450888...	-0.443008...	-0.671852...	0.434839...	0.296511...	0.729406...	0.1610...
6	P6	-0.614338...	0.032958...	-0.717007...	0.208416...	0.158817...	-0.922983...	-0.095700...	-0.887764...	0.331508...	-0.3702...
7	P7	0.090861...	-0.702202...	0.139724...	-0.543969...	0.059670...	0.989640...	0.369464...	-0.492003...	0.751755...	0.4754...
8	P8	0.335265...	0.171270...	-0.358056...	-0.579177...	0.588075...	0.677457...	-0.366356...	0.282144...	-0.773305...	-0.8401...
9	P9	-0.140485...	0.203804...	-0.484317...	0.971309...	0.739984...	-0.122900...	0.752769...	0.375253...	0.970603...	0.6716...
10	P10	0.010601...	-0.605087...	0.130913...	-0.428155...	-0.771375...	-0.324729...	0.619697...	-0.366767...	-0.568867...	0.7056...
11	C01	-0.537062...	-0.069114...	0.406454...	0.147825...	-0.101313...	-0.195468...	-0.050126...	-0.785405...	-0.009539...	-0.1145...
12	C11	-0.308369...	-0.429632...	-0.219504...	0.115538...	-0.029898...	-0.969739...	-0.698087...	-0.489413...	0.157871...	0.6584...
13	C21	-0.377276...	0.876992...	0.556554...	0.432041...	0.042760...	0.441828...	0.962499...	-0.512927...	-0.269904...	-0.5225...
14	C31	0.706618...	0.453387...	0.577065...	-0.245633...	0.084619...	0.274405...	0.946143...	-0.844618...	-0.213556...	-0.6676...
15	C41	-0.082822...	-0.402931...	-0.393328...	-0.134011...	0.431118...	0.581480...	0.638731...	-0.023047...	0.813037...	0.6680...
16	C51	-0.813339...	-0.189987...	-0.121235...	0.528789...	0.368124...	0.205085...	0.503267...	-0.181423...	0.841678...	0.4654...
17	Cm32.0	0.812282...	0.135608...	0.714368...	-0.840002...	-0.865615...	-0.212914...	-0.155278...	-0.238595...	1.032283...	-0.4475...
GENERASI 2											
1	P1	0.010798...	-0.506242...	0.130636...	-0.428417...	-0.771106...	-0.325187...	0.619509...	-0.368654...	-0.570043...	0.7047...
2	P2	0.916847...	0.893786...	0.611076...	0.971544...	0.004145...	0.068968...	1.079028...	-0.905990...	-0.694188...	-0.8388...
3	P3	-0.470979...	-0.937638...	0.897155...	0.431452...	-0.413318...	-0.724153...	0.428871...	0.292131...	0.724192...	0.1590...
4	P4	0.612340...	0.134446...	0.713755...	-0.840420...	-0.865167...	-0.213903...	-0.155196...	-0.237255...	1.033095...	-0.4466...
5	P5	0.763311...	-0.496003...	0.634993...	-0.976564...	-0.742928...	-0.479248...	-0.141213...	-0.026839...	1.093876...	-0.3212...
6	P6	-0.013363...	-0.188986...	-0.421181...	0.528808...	0.368084...	0.205162...	0.508190...	-0.181754...	0.841665...	0.4683...
7	P7	-0.307714...	-0.430178...	-0.215612...	0.013342...	-0.029222...	-0.960102...	-0.698064...	-0.499580...	0.157725...	0.6583...

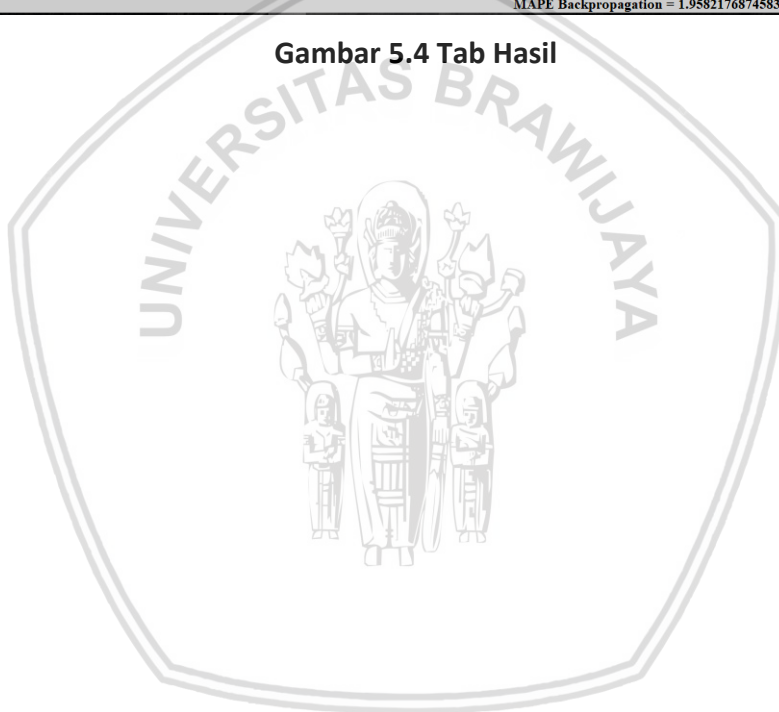
Gambar 5.3 Tab Pelatihan Algoritme Genetika - *Backpropagation*

5.4.4 Tab Pengujian dan Hasil

Pada tab pengujian dan hasil akan ditampilkan individu terbaik dari hasil pelatihan menggunakan *Backpropagation* dan individu terbaik dari hasil pelatihan menggunakan Algoritme Genetika – *Backpropagation*. Terdapat tombol lakukan pengujian pada kedua individu terpilih yang akan melakukan proses melakukan pengujian dengan menggunakan data uji yang nantinya akan menampilkan tabel hasil pengujian dan nilai MAPE dari kedua individu terpilih. Lebih lengkap dapat dilihat pada Gambar 5.4



Gambar 5.4 Tab Hasil



BAB 6 PENGUJIAN DAN ANALISIS

Bab 6 membahas pengujian pada setiap parameter pada metode *Backpropagation* dan metode Algoritme Genetika – *Backpropagation*. Hasil terbaik dari setiap parameter akan digunakan dalam validasi hasil dengan data latih dan pengujian hasil dengan data uji pada *Backpropagation* yang menggunakan inisialisasi bobot awal dengan *nguyen-widrow* dan Algoritme Genetika- *Backpropagation*.

6.1.1 Pengujian Maksimum Iterasi

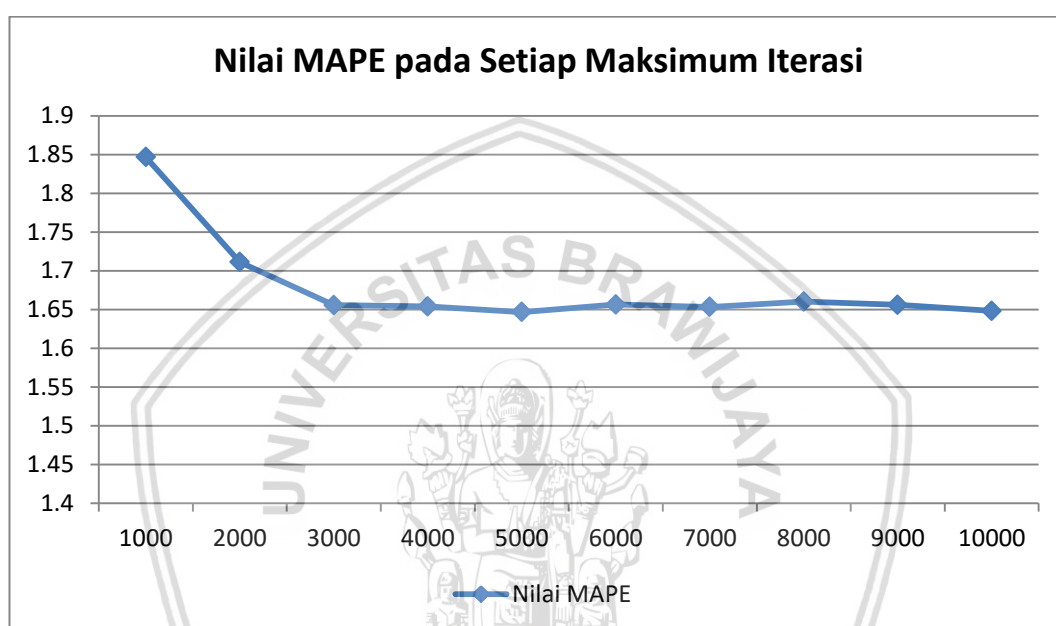
Pada pengujian ini akan dilakukan pelatihan *Backpropagation* pada maksimum iterasi 1.000 hingga 10.000 iterasi agar mendapatkan nilai maksimum iterasi yang paling optimal pada pelatihan *Backpropagation*. Nilai *learning rate* yang akan digunakan pada setiap pelatihan adalah 0.1 dan jumlah *neuron* yang digunakan adalah 5. Tabel 6.1 yang akan menampilkan hasil pengujian maksimum iterasi pada pelatihan *Backpropagation*.

Tabel 6.1 Hasil pengujian maksimum iterasi pada pelatihan *Backpropagation*

Jumlah Iterasi	Percobaan ke-										Rataan MAPE
	1	2	3	4	5	6	7	8	9	10	
1000	1,849 228 6	1,863 50279 6	1,870 059	1,808 99182	1,857 08172 5	1,836 568	1,866 914	1,827 34184 1	1,844 918	1,842 73819	1,84673 4
2000	1,743 908	1,728 25623	1,695 135	1,669 10314	1,687 63163 7	1,756 442	1,708 597	1,700 90149 6	1,716 702	1,706 16255 2	1,71128 4
3000	1,675 963	1,656 03149 1	1,642 238	1,653 58878	1,671 44114 3	1,652 606	1,652 277	1,642 66840 9	1,657 554	1,650 78137 2	1,65551 5
4000	1,648 665	1,677 93007 2	1,663 294	1,653 42856	1,619 03580 1	1,645 523	1,643 838	1,685 43823 8	1,648 295	1,654 67410 4	1,65401 2
5000	1,619 438	1,643 72757	1,651 066	1,636 65968	1,649 63660 7	1,632 238	1,648 165	1,687 43828 8	1,647 767	1,651 23642 9	1,64673 7
6000	1,651 956	1,646 37512	1,674 74	1,658 39403	1,666 88976 2	1,650 634	1,652 284	1,648 56703 4	1,660 046	1,655 36645	1,65652 5
7000	1,650 127	1,685 87305 6	1,623 866	1,656 33658	1,656 08742 8	1,643 049	1,653 226	1,647 44115 3	1,680 658	1,635 76481	1,65324 3
8000	1,643 618	1,696 43598 3	1,657 735	1,675 89138	1,645 71725 7	1,679 538	1,658 198	1,663 10391 2	1,623 856	1,656 84253 5	1,66009 4
9000	1,632 468	1,630 46192 4	1,639 759	1,692 81781	1,662 04693 6	1,685 72	1,650 124	1,686 45758 6	1,630 826	1,649 66982 6	1,65603 5

Jumlah Iterasi	Percobaan ke-										Rataan MAPE
	1	2	3	4	5	6	7	8	9	10	
10000	1,677 593	1,629 91205 6	1,607 171	1,699 29316	1,628 75320 1	1,666 691	1,644 931	1,634 31814 7	1,637 281	1,655 38538 3	1,64813 3

Agar memudahkan dalam membaca hasil pengujian nilai maksimum iterasi pada pelatihan *Backpropagation*, Gambar 6.1 menjelaskan nilai MAPE dari setiap nilai maksimum iterasi yang diujikan dalam bentuk grafik.



Gambar 6.1 Grafik hasil pengujian maksimum iterasi pada pelatihan *Backpropagation*

Dari hasil yang telah didapat pada Tabel 6.1 dan Gambar 6.1 dapat dilihat bahwa maksimum iterasi yang digunakan pada pelatihan *Backpropagation* mempengaruhi nilai *error* yang dihasilkan. Untuk nilai MAPE terbaik didapatkan dari pelatihan pada maksimum iterasi. Jumlah iterasi yang dilakukan pada *Backpropagation* menunjukkan waktu yang diperlukan *Backpropagation* untuk mencapai konvergen. Terlihat grafik cenderung membaik, yang memperlihatkan pada iterasi 1.000 dan 2.000 jumlah iterasi masih belum cukup untuk mencapai konvergen atau dapat dikatakan algoritme masih *underfitting* dalam mengenali pola dari data. Begitupun pada iterasi 6.000 hingga 10.000 yang mengalami perubahan tidak menentu walaupun kecil, hal ini dipengaruhi oleh konvergensi yang *overfitting* pada algoritme yang dilatih. Dari hasil pengujian tersebut iterasi dengan nilai MAPE terkecil adalah pada iterasi 5.000 yaitu 1,646737, maka maksimum iterasi yang sebaiknya digunakan dalam pelatihan *Backpropagation* adalah 5.000 iterasi.

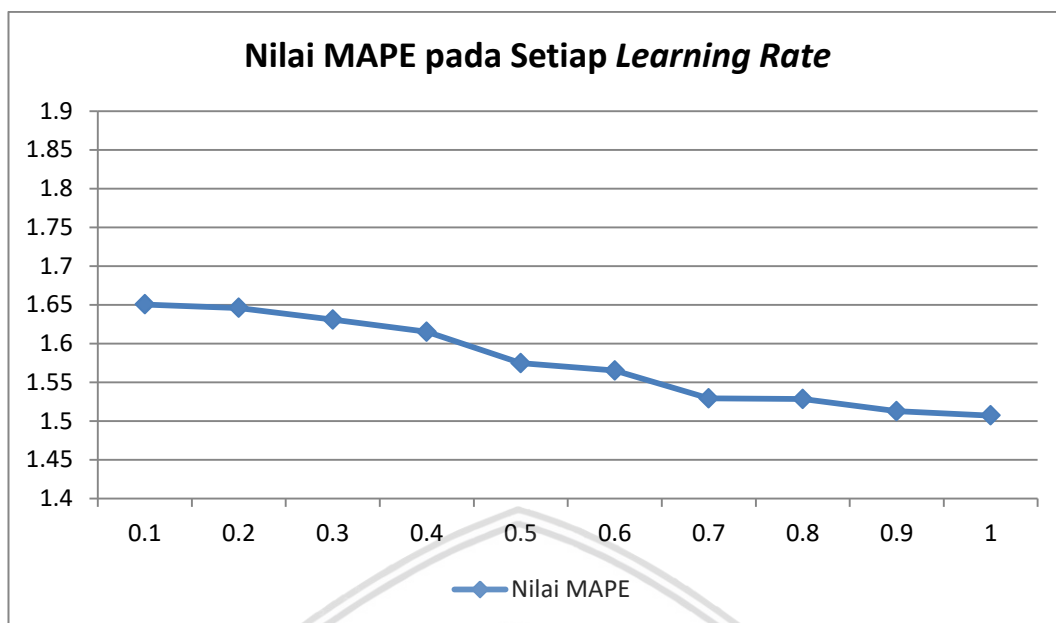
6.1.2 Pengujian Nilai *Learning Rate* (*Alpha*)

Kemudian melakukan pengujian untuk mendapatkan nilai *learning rate* yang paling optimal yang akan diujikan pada pelatihan *Backpropagation* dengan menggunakan 5 *neuron* pada lapisan tersembunyi dan jumlah iterasi sebanyak 5000 iterasi. Nilai *learning rate* yang akan diujikan adalah 0,1 hingga 1. Pada Tabel 6.2 dijelaskan hasil dari pengujian nilai *learning rate* pada rentang 0,1 hingga 1.

Tabel 6.2 Hasil pengujian nilai *learning rate* pada pelatihan *Backpropagation*

Jumlah <i>Learning Rate</i>	Percobaan ke-										Rataan MAPE
	1	2	3	4	5	6	7	8	9	10	
0,1	1,634 99	1,657 04329	1,650 278	1,645 79909	1,671 59704 9	1,645 821	1,650 174	1,625 56533 6	1,645 958	1,677 84768 6	1,65050 7
0,2	1,644 158	1,677 90707 7	1,664 446	1,628 7428	1,621 37397 4	1,636 12	1,640 013	1,637 32539 9	1,641 339	1,668 58154 7	1,64600 1
0,3	1,638 641	1,669 75012 3	1,609 053	1,632 55427	1,635 70865 6	1,626 993	1,636 633	1,654 31401 1	1,612 909	1,591 17698	1,63077 3
0,4	1,605 227	1,617 24804 7	1,587 661	1,596 34357	1,659 54965 3	1,582 038	1,613 291	1,648 76263 9	1,613 696	1,628 24827	1,61520 6
0,5	1,556 806	1,571 9284	1,542 363	1,539 89421	1,552 98212 7	1,622 419	1,625 397	1,552 85574	1,592 828	1,587 88909 2	1,57453 6
0,6	1,529 133	1,591 58501 2	1,645 132	1,600 1079	1,564 94487 5	1,583 814	1,572 993	1,513 07003 7	1,504 907	1,545 44627 4	1,56511 3
0,7	1,484 673	1,674 37758 6	1,566 025	1,501 87186	1,508 54264 6	1,503 399	1,514 798	1,537 91690 2	1,499 624	1,500 98191 8	1,52922 1
0,8	1,487 986	1,552 78318 5	1,530 452	1,650 85265	1,504 62102 1	1,494 821	1,478 257	1,510 75158 2	1,478 353	1,593 98308 6	1,52828 6
0,9	1,477 403	1,478 57501 3	1,547 595	1,512 4432	1,493 86327	1,594 07	1,482 718	1,498 91235 4	1,560 19	1,480 57654 9	1,51263 5
1	1,475 755	1,573 68524 7	1,478 748	1,477 99728	1,523 80048 6	1,524 475	1,507 367	1,545 74673 3	1,476 183	1,485 55911	1,50693 2

Agar memudahkan dalam membaca hasil pengujian *learning rate* pada pelatihan *Backpropagation*, Gambar 6.2 menjelaskan nilai MAPE dari setiap *learning rate* yang diujikan dalam bentuk grafik.



Gambar 6.2 Grafik hasil pengujian *learning rate* pada pelatihan *Backpropagation*

Dari hasil yang telah didapat pada Tabel 6.2 dan Gambar 6.2 dapat dilihat bahwa nilai *learning rate* yang digunakan pada pelatihan *Backpropagation* mempengaruhi nilai MAPE yang dihasilkan dimana hasil semakin membaik pada setiap kenaikan nilai *learning rate* yang diujikan. Untuk nilai MAPE terbaik didapatkan dari pelatihan dengan menggunakan *learning rate* 1 yaitu dengan nilai MAPE rata-rata sebesar 1,506932. Hal ini terjadi karna pelatihan yang sebenarnya masih dapat membaik pada iterasi 5.000 hanya saja dengan *learning rate* yang terlalu kecil menyebabkan perbaikan *error* pada pelatihan terlalu kecil sehingga pencapaian konvergensi yang lebih baik memerlukan iterasi yang lebih banyak meskipun pada pengujian jumlah iterasi hasil yang diberikan hingga iterasi 10.000 tidak menunjukkan perubahan yang signifikan bahkan cenderung memburuk. Dari hasil pengujian ini maka nilai *learning rate* yang sebaiknya digunakan dalam pengujian selanjutnya adalah 1.

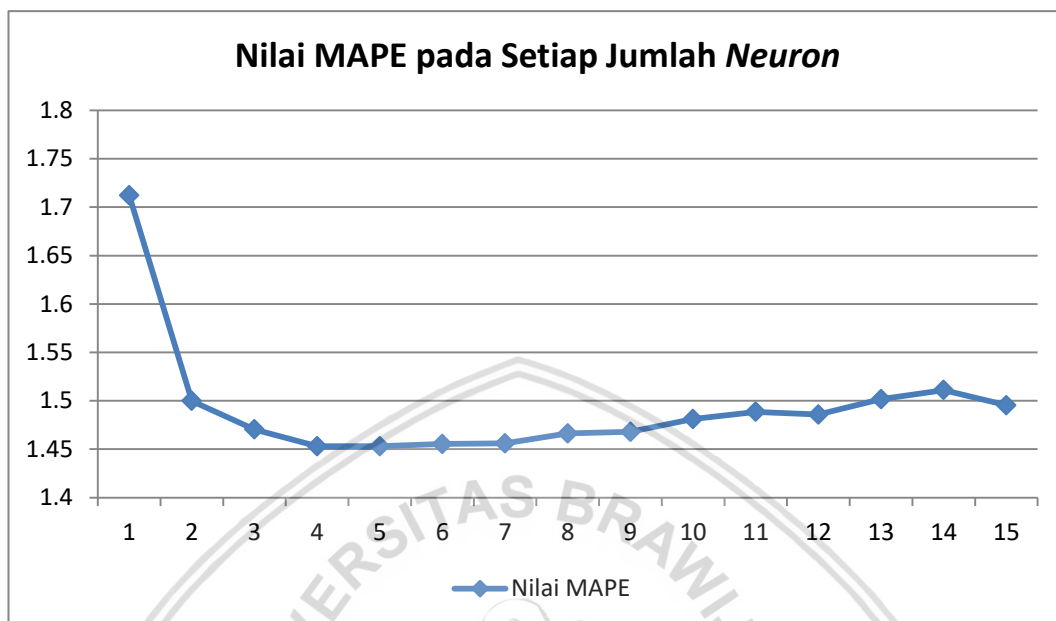
6.1.3 Pengujian Jumlah *Neuron* pada Lapisan Tersembunyi

Pada tahap ini dilakukan pengujian terhadap jumlah *neuron* pada lapisan tersembunyi untuk mendapatkan arsitektur *Backpropagation* yang paling optimal untuk melakukan prediksi. Jumlah *neuron* yang diujikan antara lain dari 1 hingga 15 jumlah *neuron* pada lapisan tersembunyi. Sedangkan jumlah iterasi yang digunakan adalah sebanyak 5.000 iterasi dan nilai *learning rate* (α) yang digunakan adalah 1. Jumlah iterasi dan *learning rate* yang digunakan sama pada semua pengujian jumlah *neuron* pada lapisan tersembunyi. Pada Tabel 6.3 ditunjukkan hasil pelatihan menggunakan *Backpropagation*.

Tabel 6.3 Hasil pengujian jumlah *neuron* pada pelatihan *Backpropagation*

Jumlah Neuron	Percobaan ke-										Rataan MAPE
	1	2	3	4	5	6	7	8	9	10	
1	1,716 496	1,711 11816 8	1,711 126	1,711 14468	1,711 12719 5	1,711 131	1,711 118	1,711 11272 6	1,716 493	1,711 13026 9	1,7122
2	1,475 377	1,503 73378 6	1,611 277	1,475 20193	1,478 24710 2	1,485 819	1,488 347	1,489 68115 8	1,506 123	1,484 63044 9	1,49984 4
3	1,476 693	1,473 25857 4	1,479 449	1,507 40785	1,474 96435	1,449 306	1,469 612	1,491 77046 7	1,409 909	1,472 17246 8	1,47045 4
4	1,464 289	1,454 75474	1,449 457	1,422 5306	1,426 37292 3	1,479 025	1,445 298	1,455 54412 6	1,469 562	1,462 94478 3	1,45297 8
5	1,532 88	1,418 36274 6	1,465 022	1,424 32644	1,426 77393 6	1,434 135	1,433 403	1,491 13842 1	1,426 726	1,476 96282 5	1,45297 3
6	1,473 268	1,448 51716 3	1,441 555	1,426 56526	1,489 99712 3	1,429 395	1,434 705	1,490 70024 1	1,453 569	1,466 07140 6	1,45543 4
7	1,447 161	1,450 55487 5	1,486 519	1,461 23791	1,458 01862 8	1,437 103	1,437 124	1,446 09343 5	1,467 685	1,469 49730 2	1,45609 9
8	1,463 52	1,449 43886	1,501 07	1,455 73288	1,492 21433 4	1,422 15	1,450 348	1,476 71411 9	1,488 393	1,463 21016 3	1,46627 9
9	1,479 624	1,476 25300 9	1,486 779	1,455 07381	1,480 77866 5	1,465 842	1,480 943	1,472 02770 5	1,440 801	1,440 75487 8	1,46788 8
10	1,481 768	1,508 57850 4	1,488 05	1,449 38247	1,504 59963 3	1,477 067	1,496 729	1,471 26220 7	1,445 449	1,489 40714 5	1,48122 9
11	1,484 133	1,512 51631 7	1,495 285	1,480 90209	1,473 69168 8	1,484 195	1,502 34	1,490 12561 2	1,472 701	1,490 77735 2	1,48866 7
12	1,478 405	1,452 86658 4	1,466 667	1,459 70188	1,499 07348	1,470 848	1,505 578	1,519 78239 3	1,508 793	1,496 78656 9	1,48585
13	1,490 931	1,542 09479 6	1,491 459	1,525 72741	1,467 00368 5	1,498 307	1,506 48	1,498 94885 1	1,507 596	1,488 56425 7	1,50171 1
14	1,507 509	1,520 09987 9	1,513 703	1,518 03738	1,492 21838 3	1,477 8	1,533 899	1,508 96429 4	1,510 136	1,528 94379 8	1,51113 1
15	1,489 324	1,503 55248 9	1,520 936	1,493 80871	1,460 90694 3	1,474 741	1,501 961	1,506 83290 3	1,518 255	1,482 97591 4	1,49532 9

Agar memudahkan dalam membaca hasil pelatihan, Gambar 6.3 menjelaskan nilai MAPE dari setiap jumlah *neuron* pada lapisan tersembunyi dalam bentuk grafik.



Gambar 6.3 Grafik hasil pengujian jumlah *neuron* pada pelatihan *Backpropagation*

Dari hasil yang telah didapat pada Tabel 6.3 dan Gambar 6.3 dapat dilihat bahwa jumlah *neuron* yang digunakan pada arsitektur *Backpropagation* mempengaruhi nilai MAPE yang dihasilkan. Pada arsitektur dengan jumlah *neuron* yang terlalu sedikit membuat hasil kurang optimum, hal ini disebabkan jumlah *neuron* yang terlalu sedikit pada lapisan tersembunyi kurang mampu memberikan fungsi matematis yang sesuai untuk mengenali pola pada data jika data terlalu kompleks sehingga *error* yang dihasilkan masih terlalu besar atau dapat disimpulkan bahwa pada *neuron* yang terlalu sedikit terjadi *underfitting* dimana algoritme masih terlalu samar dalam mengenali pola pada data. Untuk nilai MAPE terkecil didapatkan dari pelatihan dengan menggunakan 5 *neuron* pada lapisan tersembunyi dengan rata-rata MAPE sebesar 1,452973. Sedangkan pada jumlah *neuron* yang lebih tinggi MAPE yang dihasilkan mengalami kenaikan, hal ini terjadi karna pada jumlah *neuron* yang semakin banyak fungsi matematis yang terbentukpun semakin kompleks sehingga algoritme semakin sulit konvergen pada solusi yang optimum. Dari hasil pengujian tersebut maka jumlah *neuron* yang sebaiknya digunakan dalam pengujian selanjutnya adalah *Backpropagation* dengan 5 *neuron* pada lapisan tersembunyi.

6.2 Validasi dan Pengujian *Backpropagation*

Validasi dan Pengujian pada metode *Backpropagation* bertujuan untuk mendapatkan nilai akurasi pelatihan *Backpropagation* terhadap data latih dan data uji dengan menggunakan parameter yang mampu memberikan hasil

optimum pada pengujian parameter sebelumnya. Validasi dilakukan dengan menggunakan hasil bobot terbaik hasil pelatihan sebagai bobot untuk memprediksi data latih. Sedangkan pengujian dilakukan dengan menggunakan hasil bobot terbaik dari pelatihan menggunakan *Backpropagation* terhadap data uji. Validasi dan Pengujian dilakukan dengan menggunakan 5 *neuron* pada lapisan tersembunyi, 1 pada *learning rate* dan 5.000 iterasi. Tabel menunjukkan hasil pengujian akurasi pada pelatihan *Backpropagation*.

Tabel 6.4 Hasil validasi dan pengujian akurasi pada pelatihan *Backpropagation*

Percobaan	MAPE Validasi	MAPE Pengujian
1	1,434163	1,869233
2	1,413194	1,854375
3	1,44981	1,911378
4	1,454306	1,89774
5	1,421356	1,891141
6	1,428243	1,895443
7	1,481931	1,673873
8	1,516604	1,742582
9	1,471335	1,908318
10	1,419833	1,874848
Rata-Rata	1,449077	1,851893

Tabel 6.4 menunjukkan hasil yang baik pada validasi dan pengujian, meskipun pada pengujian tidak memberikan hasil sebaik validasi namun nilai MAPE ini masih cukup kecil. Dengan rata-rata MAPE pada validasi sebesar 1.449077 dan rata-rata MAPE pada pengujian sebesar 1,851893.

6.3 Pelatihan Algoritme Genetika – *Backpropagation*

6.3.1 Pengujian Jumlah Generasi

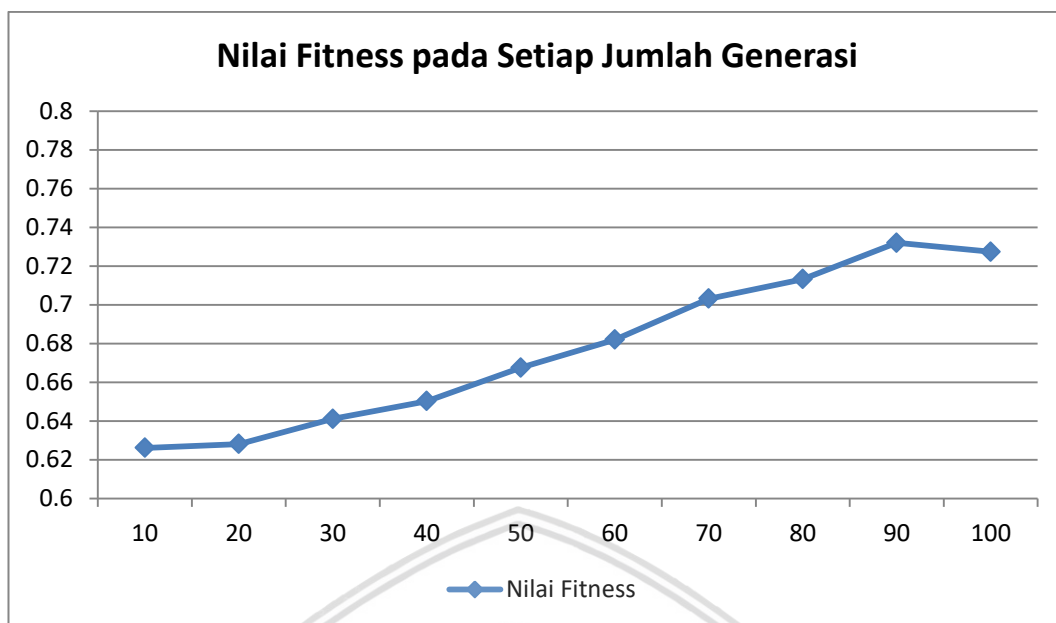
Pengujian jumlah Generasi bertujuan untuk mendapatkan jumlah generasi yang memberikan hasil paling optimal dalam melakukan pelatihan Algoritme Genetika – *Backpropagation*. Pengujian akan dilakukan sebanyak 10 kali dengan jumlah generasi mulai 10 generasi hingga 100 generasi. Pengujian akan dilakukan dengan ukuran populasi sebesar 50, *crossover rate* 0,6, *mutation rate* 0,001, jumlah *neuron* 5, *learning rate* 1, dan 100 jumlah iterasi pada pelatihan *Backpropagation*. Setiap pengujian dilakukan sebanyak 10 kali dan kemudian diambil nilai rata-ratanya. Hasil pengujian dijabarkan pada Tabel 6.5

**Tabel 6.5 Hasil pengujian jumlah generasi pada pelatihan Algoritme Genetika -
Backpropagation**

Jumlah Generasi	Percobaan ke-										Rataan <i>fitness</i>
	1	2	3	4	5	6	7	8	9	10	
10	0,636 746	0,623 64947	0,627 2478	0,624 495	0,627 11	0,623 816	0,629 436	0,618 987	0,625 448	0,624 295	0,62612 3
20	0,624 746	0,632 00492	0,623 9915	0,625 315	0,635 299	0,632 325	0,627 156	0,626 746	0,627 761	0,625 969	0,62813 1
30	0,632 478	0,650 54355	0,655 0937	0,639 236	0,621 445	0,619 507	0,637 477	0,637 71	0,662 057	0,655 506	0,64110 5
40	0,646 749	0,647 46867	0,654 0567	0,625 234	0,667 232	0,641 854	0,658 096	0,663 581	0,638 166	0,659 998	0,65024 4
50	0,685 149	0,647 60873	0,674 4131	0,686 051	0,665 322	0,665 603	0,648 672	0,658 577	0,651 612	0,691 862	0,66748 7
60	0,674 954	0,703 71809	0,692 7762	0,691 075	0,687 632	0,659 315	0,655 618	0,663 355	0,704 525	0,687 246	0,68202 1
70	0,691 49	0,699 44591	0,698 8	0,706 585	0,708 512	0,705 385	0,714 155	0,718 749	0,693 524	0,695 068	0,70317 1
80	0,734 716	0,692 85977	0,692 5498	0,709 904	0,720 621	0,726 383	0,690 892	0,716 469	0,721 343	0,726 76	0,71325
90	0,740 366	0,730 11784	0,741 689	0,728 534	0,731 478	0,735 188	0,729 322	0,729 57	0,731 476	0,722 479	0,73202 2
100	0,714 982	0,736 88706	0,714 9332	0,742 09	0,724 337	0,706 417	0,742 358	0,741 267	0,714 539	0,736 123	0,72739 3

Hasil pengujian yang telah dijabarkan dalam Tabel 6.5 digambarkan dalam bentuk grafik pada Gambar 6.4

Pada Gambar 6.4 menunjukkan bahwa jumlah populasi akan mempengaruhi hasil yang diberikan dari proses pelatihan. Hasil pada proses pelatihan relative membaik pada setiap peningkatan jumlah generasi. Hasil yang semakin membaik menunjukkan bahwa eksplorasi yang dilakukan Algoritme Genetika berhasil memberikan solusi yang lebih optimum bagi *Backpropagation* dengan terpilihnya individu yang lebih baik dari setiap proses seleksi pada setiap generasi. Generasi terbaik berada pada 90 generasi dengan nilai rataan *fitness* mencapai 0,732022. Sedangkan pada generasi 100 terjadi penurunan nilai *fitness* yang menunjukkan Algoritme Genetika sudah mencapai titik konvergen pada generasi sebelumnya. Dari hasil tersebut maka banyak generasi yang akan digunakan pada pengujian selanjutnya adalah 90 generasi.



Gambar 6.4 Grafik Hasil pengujian jumlah Generasi pada pelatihan Algoritme Genetika – *Backpropagation*

6.3.2 Pengujian Ukuran Populasi

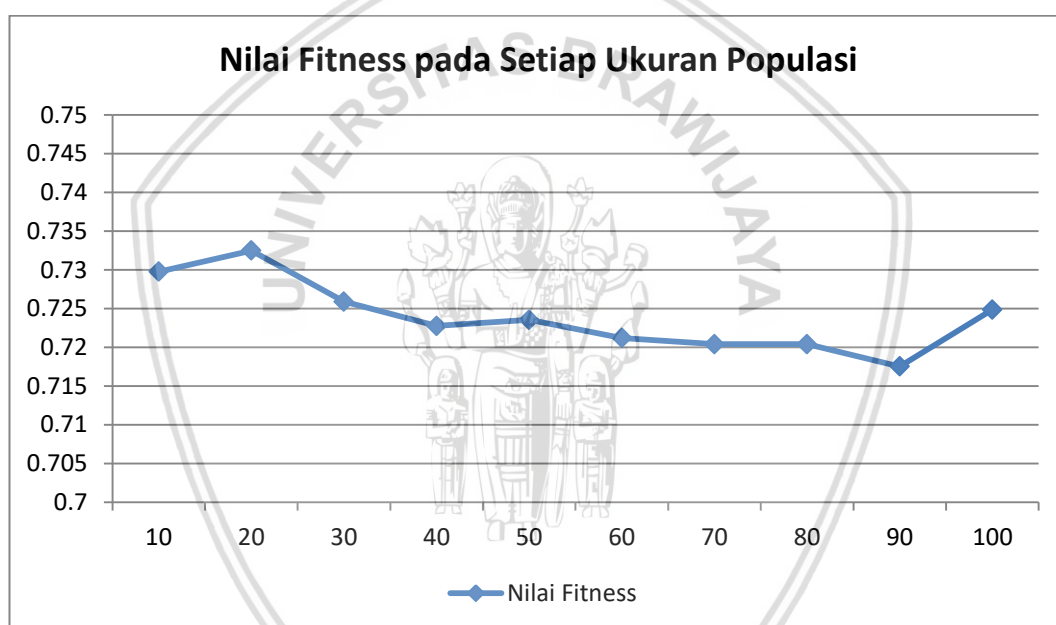
Pengujian ukuran populasi dilakukan dengan tujuan agar mendapatkan ukuran populasi yang mampu memberikan hasil paling maksimal pada pelatihan Algoritme Genetika – *Backpropagation*. Pada pengujian ukuran populasi akan dilakukan percobaan sebanyak sepuluh kali pada setiap ukuran populasi. Ukuran populasi yang diujikan adalah 10 hingga 100. Pengujian akan dilakukan dengan menggunakan generasi sebesar 90 generasi, 0,6 *crossover rate*, 0,001 *mutation rate*, 5 *neuron* pada lapisan tersembunyi, 1 *learning rate*, dan 100 iterasi pada pelatihan *Backpropagation* di dalam Algoritme Genetika. Hasil pengujian ditampilkan dalam Tabel 6.6.

Tabel 6.6 Hasil pengujian ukuran populasi pada pelatihan Algoritme Genetika - *Backpropagation*

Ukuran Populasi	Percobaan ke-										Rataan <i>fitness</i>
	1	2	3	4	5	6	7	8	9	10	
10	0,730 516	0,728 68791	0,739 4304	0,726 159	0,713 936	0,730 924	0,731 809	0,716 766	0,740 465	0,738 962	0,72976 6
20	0,723 48	0,746 19378	0,737 6367	0,734 978	0,739 42	0,722 315	0,731 626	0,724 199	0,740 925	0,724 133	0,73249 1
30	0,740 58	0,720 10017	0,724 5186	0,730 25	0,734 145	0,726 161	0,743 468	0,712 908	0,705 177	0,721 643	0,72589 5
40	0,731 103	0,722 17415	0,723 3413	0,721 781	0,724 409	0,728 009	0,737 039	0,695 622	0,725 464	0,718 661	0,72276
50	0,718 58	0,737 50688	0,702 4054	0,723 076	0,727 553	0,737 425	0,737 118	0,699 738	0,718 362	0,733 699	0,72354 6

Ukuran Populasi	Percobaan ke-										Rataan <i>fitness</i>
	1	2	3	4	5	6	7	8	9	10	
60	0,724 686	0,712 18047	0,735 722	0,716 193	0,739 871	0,688 829	0,731 898	0,704 142	0,730 448	0,727 953	0,72119 2
70	0,725 454	0,712 51942	0,717 9523	0,735 035	0,722 895	0,723 341	0,713 557	0,735 22	0,738 778	0,679 15	0,72039
80	0,731 265	0,735 48266	0,730 4827	0,741 006	0,707 273	0,709 83	0,704 335	0,722 155	0,712 252	0,709 948	0,72040 3
90	0,701 12	0,710 48178	0,722 7297	0,724 504	0,690 192	0,728 591	0,712 887	0,731 371	0,725 175	0,728 155	0,71752 1
100	0,712 98	0,742 30515	0,718 6837	0,721 578	0,720 126	0,724 44	0,732 755	0,719 013	0,729 558	0,727 155	0,72485 9

Agar memudahkan dalam membaca hasil pengujian jumlah generasi, hasil juga ditampilkan dalam bentuk grafik pada Gambar 6.5



Gambar 6.5 Grafik Hasil pengujian ukuran populasi pada pelatihan Algoritme Genetika – *Backpropagation*

Gambar 6.5 menunjukkan hasil yang relative sama dengan *fitness* yang cenderung berada di sekitar 0,72 hingga 0,73. Hasil cenderung memburuk pada setiap kenaikan nilai ukuran populasi, hal ini dapat terjadi karena semakin besar ukuran populasi yang digunakan semakin sulit individu terbaik untuk dapat terus lolos pada setiap generasinya jika mengingat jenis seleksi yang digunakan adalah *Roulette Wheel* yang masih didasari oleh probabilitas pemilihan meskipun nilai probabilitas didapatkan berdasarkan nilai *fitness*-nya. Hasil terbaik berada pada ukuran populasi 20 dengan *fitness* mencapai 0,732491. Dengan begitu ukuran populasi 20 yang akan digunakan pada pengujian selanjutnya.

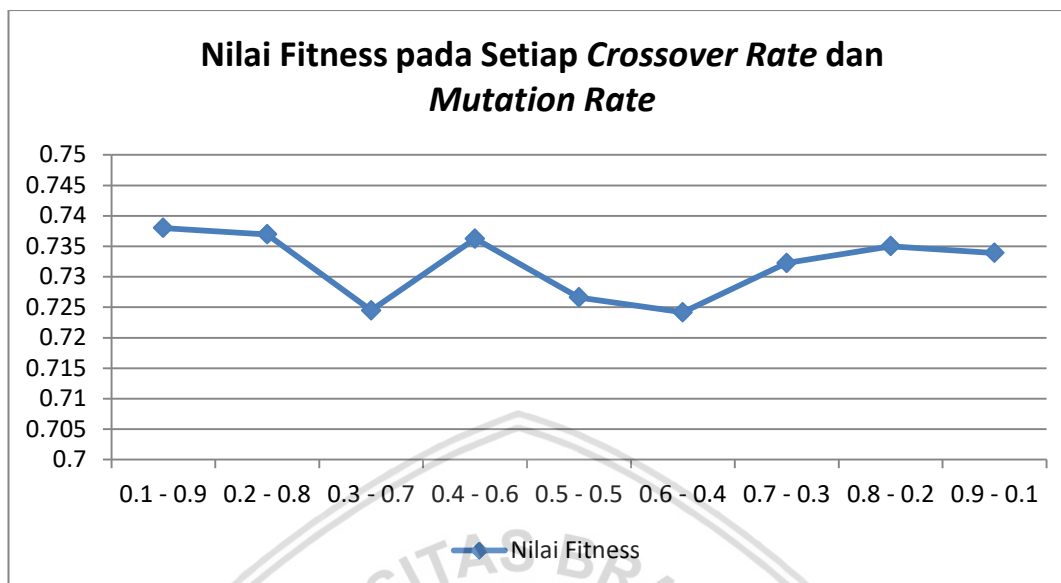
6.3.3 Pengujian *Crossover Rate* dan *Mutation Rate*

Pengujian *crossover rate*(cr) dan *mutation rate*(mr) dilakukan dengan kombinasi cr dan mr 0,1-0,9, 0,2-0,8, 0,3-0,7, 0,4-0,6, 0,5-0,5, 0,6-0,4, 0,7-0,3, 0,8-0,2, 0,9-0,1 Pengujian dilakukan dalam sepuluh kali percobaan. Pengujian akan dilakukan dengan menggunakan generasi sebesar 90 generasi, 20 ukuran populasi, 5 *neuron* pada lapisan tersembunyi, 1 *learning rate*, dan 100 iterasi pada pelatihan *Backpropagation* di dalam Algoritme Genetika Hasil pengujian ditunjukkan pada Tabel 6.7.

Tabel 6.7 Hasil pengujian *crossover rate* dan *mutation rate* pada pelatihan Algoritme Genetika - *Backpropagation*

rate		Percobaan ke-										Rataan Fitness
cr	mr	1	2	3	4	5	6	7	8	9	10	
0,1	0,9	0,724 502	0,760 0377	0,701 0599	0,742 705	0,733 616	0,749 367	0,742 372	0,738 577	0,754 785	0,733 024	0,738005
0,2	0,8	0,735 685	0,721 4858 2	0,736 681	0,737 299	0,708 711	0,740 548	0,746 357	0,754 397	0,756 595	0,731 796	0,736955
0,3	0,7	0,706 395	0,733 8335 7	0,725 3618	0,723 156	0,728 576	0,733 201	0,714 103	0,723 589	0,721 829	0,734 757	0,72448
0,4	0,6	0,755 448	0,746 7147 3	0,736 8578	0,734 148	0,733 881	0,738 874	0,738 187	0,734 791	0,727 873	0,715 757	0,736253
0,5	0,5	0,723 227	0,740 0243 7	0,700 8017	0,724 338	0,733 306	0,725 742	0,749 817	0,727 089	0,709 799	0,731 93	0,726607
0,6	0,4	0,733 596	0,708 0894 8	0,688 8808	0,720 251	0,733 637	0,709 468	0,751 301	0,732 962	0,729 433	0,733 892	0,724151
0,7	0,3	0,721 18	0,734 3999 4	0,732 7702	0,723 114	0,723 735	0,740 565	0,745 692	0,743 574	0,726 851	0,730 679	0,732256
0,8	0,2	0,744 527	0,746 0221 9	0,727 8915	0,745 341	0,733 143	0,707 292	0,740 99	0,742 97	0,736 373	0,725 654	0,73502
0,9	0,1	0,725 389	0,742 8233 3	0,734 4788	0,735 581	0,727 888	0,721 941	0,738 173	0,740 701	0,734 349	0,737 693	0,733902

Agar memudahkan dalam membaca hasil pengujian, maka ditampilkan pula hasil dalam bentuk grafik seperti pada Gambar 6.6



Gambar 6.6 Grafik hasil pengujian *crossover rate* dan *mutation rate* pada pelatihan Algoritme Genetika – *Backpropagation*

Hasil dari Gambar 6.6 menunjukkan hasil yang beragam, dapat dilihat nilai berada pada kisaran 0,72 hingga 0,74. Hasil terbaik berada pada *crossover rate* 0,1 dan *mutation rate* 0,9 dan semakin memburuk pada *crossover rate* 0,3 dan *mutation rate* 0,7 hingga akhirnya membaik lagi. Nilai *crossover rate* yang relative kecil dan *mutation rate* yang besar menunjukkan Algoritma Genetika berperan sangat baik sebagai algoritme yang melakukan eksplorasi calon hasil optimum bagi *Backpropagation* tanpa harus banyak melakukan eksploitasi yang mendalam. Pada *crossover rate* yang relative tinggi dan *mutation rate* yang relative rendah nilai *fitness* yang didapatkan juga relative baik namun jika dibandingkan dengan penggunaan *crossover rate* yang relative kecil dan *mutation rate* yang relative besar nilai *fitness* yang diberikan masih kalah baik.

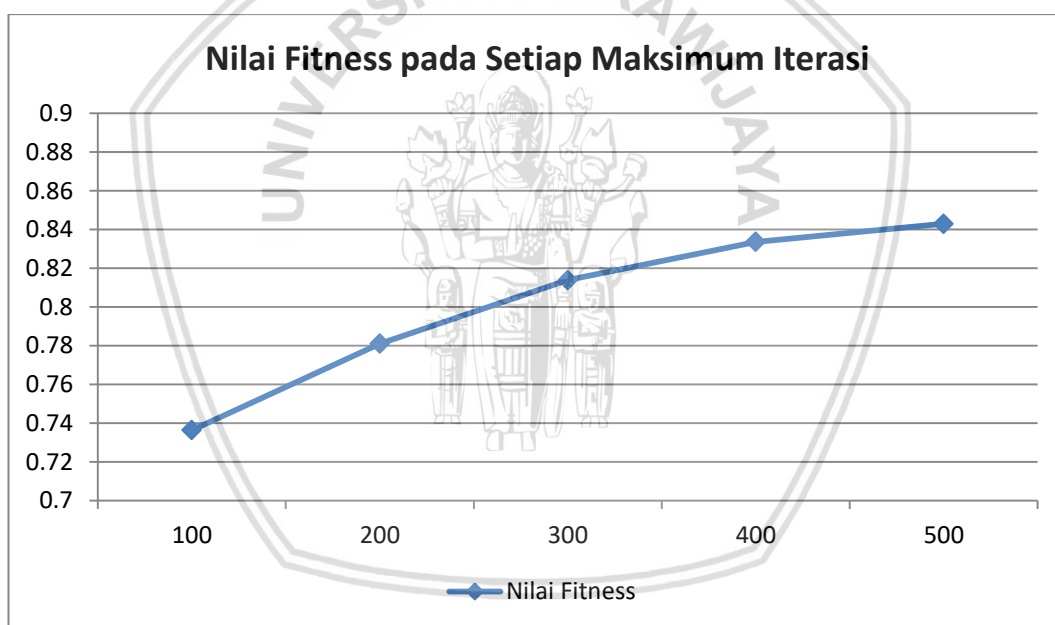
6.3.4 Pengujian Maksimum Iterasi

Pada pengujian maksimum iterasi dilakukan dengan melakukan pelatihan sebanyak sepuluh kali untuk setiap maksimum iterasi yang berbeda. Maksimum iterasi yang diujikan adalah 100 hingga 500 maksimum iterasi, dengan kenaikan setiap perubahan nilai parameter sebesar 100. Pengujian maksimum iterasi dilakukan dengan tujuan mendapatkan nilai parameter maksimum iterasi yang mampu memberikan hasil yang paling optimal. Pengujian akan dilakukan dengan 90 kali generasi, 20 ukuran populasi, 0,1 nilai *crossover rate*, 0,9 nilai *mutation rate*, 5 *neuron* pada lapisan tersembunyi, dan 0,1 nilai *learning rate*. Hasil pengujian ditampilkan pada Tabel 6.8.

Tabel 6.8 hasil pengujian maksimum iterasi pada pelatihan Algoritme Genetika - *Backpropagation*

Maksimum Iterasi	Percobaan ke-										Rataan fitness
	1	2	3	4	5	6	7	8	9	10	
100	0,768 19	0,749 64209	0,725 7985	0,748 057	0,732 984	0,722 835	0,742 194	0,748 195	0,715 745	0,709 624	0,73632 6
200	0,792 887	0,807 60002	0,744 8823	0,773 165	0,739 841	0,784 971	0,827 943	0,784 137	0,787 709	0,767 249	0,78103 8
300	0,831 348	0,817 07496	0,815 0141	0,810 32	0,814 305	0,805 983	0,827 601	0,830 874	0,774 432	0,811 503	0,81384 6
400	0,822 835	0,865 86539	0,811 919	0,820 766	0,906 65	0,793 24	0,859 284	0,821 84	0,817 378	0,815 782	0,83355 6
500	0,818 437	0,856 44146	0,858 9973	0,849 743	0,808 108	0,861 606	0,864 405	0,830 465	0,846 988	0,833 712	0,84289

Untuk memudahkan dalam membaca hasil pengujian maksimum iterasi pada Tabel 6.9 hasil ditampilkan dalam bentuk grafik pada Gambar 6.7.



Gambar 6.7 Pengujian maksimum iterasi pada pelatihan Algoritme Genetika - *Backpropagation*

Hasil yang telah didapat pada pengujian maksimum iterasi pada pelatihan Algoritme Genetika – *Backpropagation* terus membaik pada setiap kenaikan jumlah iterasi, namun pengujian dihentikan pada 500 iterasi karena jika lebih dari itu waktu komputasi yang dilakukan untuk proses pelatihan terlampau lama dan semakin kecilnya perubahan pada nilai *fitness* yang diberikan. Gambar 6.7 menunjukkan hasil yang terus membaik karena semakin banyak iterasi yang dilakukan pada *Backpropagation* semakin dalam pembelajaran terhadap data yang dapat dilakukan jika mengingat perbaikan *error* yang dilakukan pada *Backpropagation* menggunakan pendekatan *steepest descent*. Hasil terbaik

ditunjukkan pengujian dengan menggunakan 500 maksimum iterasi dengan MAPE sebesar 0,84289.

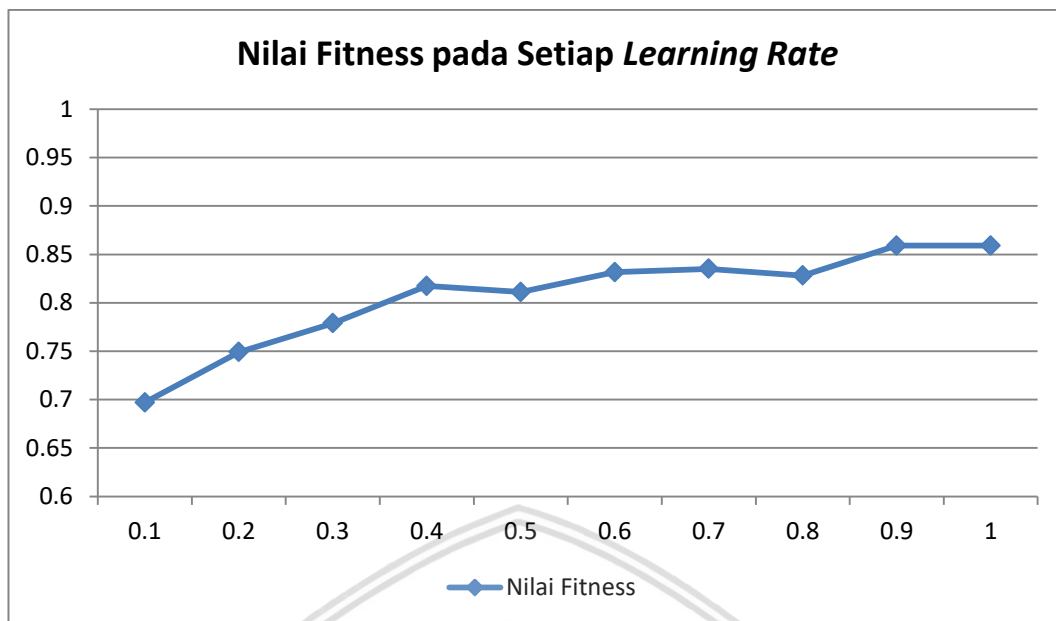
6.3.5 Pengujian *Learning Rate* (*Alpha*)

Pada pengujian *learning rate* dilakukan untuk mendapatkan nilai *learning rate* yang mampu memberikan hasil paling optimal. Pengujian akan dilakukan pada *learning rate* 0.1 hingga 1. Setiap *learning rate* diuji sebanyak sepuluh kali. Pengujian dilakukan dengan 90 kali generasi, 20 ukuran populasi, 0,1 *crossover rate*, 0,9 *mutation rate*, dan 5 *neuron* pada lapisan tersembunyi serta 500 iterasi. Hasil pengujian *learning rate* ditampilkan dalam Tabel 6.9.

Tabel 6.9 Pengujian *learning rate* pada pelatihan Algoritme Genetika - *Backpropagation*

<i>Learning Rate</i>	Percobaan ke-										Rataan <i>fitness</i>
	1	2	3	4	5	6	7	8	9	10	
0,1	0,714 513	0,698 48794	0,716 8509	0,665 464	0,700 155	0,678 351	0,701 312	0,708 249	0,711 918	0,675 605	0,69709 1
0,2	0,748 388	0,727 21275	0,758 0784	0,777 212	0,764 132	0,730 709	0,768 153	0,757 318	0,737 279	0,722 854	0,74913 4
0,3	0,809 002	0,787 60232	0,763 6076	0,780 307	0,767 709	0,777 559	0,744 167	0,754 562	0,810 837	0,793 813	0,77891 7
0,4	0,819 752	0,789 73237	0,822 2477	0,824 794	0,838 519	0,832 259	0,828 493	0,782 003	0,836 807	0,798 666	0,81732 7
0,5	0,801 534	0,812 80723	0,834 1854	0,787 54	0,788 645	0,836 227	0,777 441	0,813 563	0,812 911	0,845 722	0,81105 8
0,6	0,836 177	0,834 60394	0,787 8832	0,829 28	0,848 779	0,822 699	0,854 055	0,833 664	0,842 894	0,825 763	0,83158
0,7	0,856 077	0,814 0949	0,839 2583	0,830 142	0,867 964	0,855 773	0,786 441	0,820 658	0,846 252	0,834 652	0,83513 1
0,8	0,833 944	0,819 35813	0,845 8763	0,803 392	0,814 748	0,836 258	0,832 273	0,807 113	0,821 344	0,866 406	0,82807 1
0,9	0,918 484	0,886 67781	0,811 0458	0,846 192	0,829 176	0,866 794	0,900 949	0,832 98	0,889 831	0,807 771	0,85899
1	0,870 608	0,838 08226	0,902 8091	0,800 601	0,827 406	0,895 492	0,826 648	0,892 355	0,843 51	0,893 327	0,85908 4

Pada pengujian yang telah dilakukan dengan hasil seperti pada Tabel 6.9 agar memudahkan dalam membaca hasil pengujian, hasil pengujian ditampilkan dalam grafik pada Gambar 6.8.



Gambar 6.8 Pengujian *learning rate* pada pelatihan Algoritme Genetika - *Backpropagation*

Pada Gambar 6.8 menunjukkan hasil yang relative semakin baik dengan puncaknya pada *learning rate* 1. Hal ini terjadi karena pelatihan yang sebenarnya masih dapat membaik dengan parameter pelatihan yang ada namun pada *learning rate* 0,1 perubahan yang diberikan terlalu kecil sehingga membutuhkan iterasi pada pelatihan *Backpropagation* dan generasi pada Algoritme Genetika yang semakin besar namun jika begitu akan memerlukan waktu komputasi yang lebih lama. Dari hasil pengujian *learning rate* maka pengujian selanjutnya akan menggunakan *learning rate* 1.

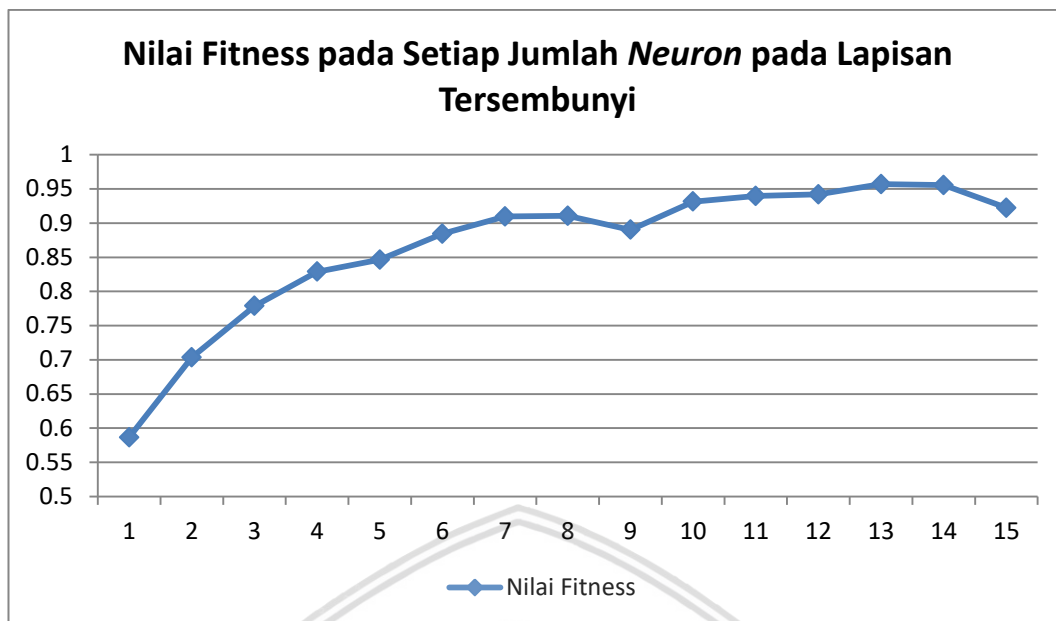
6.3.6 Pengujian Jumlah *Neuron* pada Lapisan Tersembunyi

Pada pengujian jumlah *neuron* pada lapisan tersembunyi dilakukan untuk mendapatkan jumlah *neuron* yang mampu memberikan hasil paling optimal. Jumlah *neuron* yang akan diujikan adalah 1 hingga 15 *neuron* pada lapisan tersembunyi. Setiap jumlah *neuron* akan diujikan sebanyak sepuluh kali. Pada pengujian ini akan digunakan 90 kali generasi, 20 ukuran populasi, 0,1 *crossover rate*, 0,9 *mutation rate*. Tabel 6.10 menunjukkan hasil pengujian terhadap jumlah *neuron* pada lapisan tersembunyi.

**Tabel 6.10 Pengujian jumlah *neuron* pada lapisan tersembunyi pada pelatihan
Algoritme Genetika - *Backpropagation***

Jumlah Neuron	Percobaan ke-										Rataan <i>fitness</i>
	1	2	3	4	5	6	7	8	9	10	
1	0,588 791	0,584 94244	0,585 5259	0,586 814	0,585 808	0,585 104	0,588 791	0,585 253	0,588 791	0,584 665	0,58644 9
2	0,703 763	0,706 68455	0,703 7202	0,704 791	0,704 093	0,701 785	0,703 772	0,703 942	0,700 154	0,703 11	0,70358 1
3	0,775 104	0,799 70673	0,786 0127	0,755 701	0,782 428	0,800 9	0,771 21	0,805 776	0,762 067	0,748 076	0,77869 8
4	0,771 421	0,824 68746	0,836 0433	0,822 714	0,850 021	0,833 222	0,822 244	0,828 295	0,840 519	0,858 572	0,82877 4
5	0,841 355	0,911 16283	0,863 3104	0,831 448	0,826 218	0,816 926	0,855 132	0,777 207	0,867 617	0,872 817	0,84631 9
6	0,854 063	0,859 83943	0,896 8476	0,922 046	0,952 189	0,875 846	0,900 176	0,841 672	0,858 766	0,882 597	0,88440 4
7	0,853 156	0,798 92956	0,856 7263	0,984 887	0,956 174	0,974 692	0,922 806	0,911 572	0,921 334	0,915 182	0,90954 6
8	0,930 983	0,926 58538	0,958 6998	0,872 257	0,913 012	0,922 909	0,903 848	0,878 683	0,899 649	0,899 505	0,91061 3
9	0,818 378	0,840 32468	0,906 6716	0,931 261	0,902 057	0,886 779	0,907 081	0,895 362	0,892 428	0,922 383	0,89027 3
10	1,019 51	0,854 98215	0,928 7375	0,913 769	0,988 583	0,957 465	0,924 51	0,889 78	0,934 838	0,902 253	0,93144 3
11	1,010 725	0,912 17856	0,949 399	0,905 303	0,922 844	1,000 884	0,927 335	0,914 589	0,902 49	0,951 58	0,93973 3
12	0,920 535	0,901 0681	1,058 2835	0,924 13	0,897 001	0,946 246	0,963 591	0,926 313	0,981 537	0,901 601	0,94203 1
13	0,971 051	0,981 27698	0,991 3603	0,985 613	0,968 399	0,895 298	1,010 669	0,864 985	0,963 472	0,937 859	0,95699 9
14	0,941 024	0,913 7178	0,938 5859	0,975 332	0,954 807	0,917 37	0,957 391	0,970 644	0,894 516	1,091 999	0,95553 9
15	0,947 275	0,937 87657	0,909 6718	0,927 167	0,952 228	0,918 538	0,908 509	0,867 288	0,910 059	0,943 914	0,92225 3

Dari hasil pada Tabel 6.10 akan ditampilkan hasil pengujian dalam bentuk grafik seperti pada Gambar 6.9



Gambar 6.9 Grafik pengujian jumlah *neuron* pada lapisan tersembunyi pada pelatihan Algoritme Genetika - *Backpropagation*

Hasil dalam grafik pada Gambar 6.9 menunjukkan *fitness* yang cenderung membaik pada setiap kenaikan jumlah *neuron* pada lapisan tersembunyi dengan *fitness* terbaik didapat pada jumlah *neuron* 13. Hasil tersebut menunjukkan semakin banyak jumlah *neuron* pada lapisan tersembunyi semakin baik hasil yang diberikan. Hal ini berbeda dengan hasil yang ditunjukkan pada saat pengujian jumlah *neuron* pada pelatihan *Backpropagation* saja, dimana pada jumlah *neuron* yang besar memberikan hasil yang kurang baik, hal ini sangat mungkin terjadi mengingat semakin banyak jumlah *neuron* yang digunakan maka semakin kompleks fungsi matematis yang dibentuk. Keterbatasan *Backpropagation* dalam melakukan eksplorasi calon hasil terbaik membuat *Backpropagation* kesulitan mendapatkan hasil optimum pada jumlah *neuron* yang terbilang tinggi. Dengan hasil ini pengujian pada tahap berikutnya akan menggunakan jumlah *neuron* 13 pada lapisan tersembunyi pada pelatihan Algoritme Genetika – *Backpropagation*.

6.4 Validasi dan Pengujian Algoritme Genetika – *Backpropagation*

Validasi dan pengujian ini dilakukan untuk mendapatkan nilai akurasi pada pelatihan Algoritme Genetika – *Backpropagation*. Pada validasi hasil terbaik dari pelatihan diujikan kedalam data latih, sedangkan pada pengujian diujikan kedalam data uji. Pengujian akan dilakukan sebanyak sepuluh kali dengan parameter yang sama pada setiap pengujian. Parameter yang digunakan pada pengujian adalah parameter terbaik hasil pengujian sebelumnya yaitu, jumlah generasi 90, ukuran populasi 20, *crossover rate* 0,1, *mutation rate* 0,9, jumlah *neuron* pada lapisan tersembunyi sebanyak 13, nilai *learning rate* 1, dan jumlah itersi sebanyak 500. Hasil pengujian ditunjukkan Tabel 6.11.

Tabel 6.11 Hasil validasi dan pengujian akurasi pada pelatihan Algoritme Genetika – *Backpropagation*

Percobaan	MAPE Validasi	MAPE Pengujian
1	1,0552395	1,753515
2	1,0511154	1,68927
3	1,158695	1,715814
4	1,107816	1,763197
5	1,0004917	1,780312
6	1,0902536	1,717151
7	1,096619	1,924085
8	1,0527724	1,75873
9	1,0631276	1,740078
10	1,0998978	1,575318
Rata-Rata	1,077603	1,741747

Tabel 6.11 menunjukkan hasil yang sangat baik dari validasi pada pelatihan Algoritme Genetika – *Backpropagation* dengan rata-rata MAPE sebesar 1,077603 sedangkan pada pengujian menggunakan data uji hasil yang diberikan tidak berbeda jauh dengan nilai MAPE yang dihasilkan dari pelatihan *Backpropagation* yaitu dengan rata-rata MAPE sebesar 1,741747.

6.5 Perbandingan Hasil Pengujian

Pada kedua metode yaitu *Backpropagation* maupaun Algoritma Genetika – *Backpropagation* telah dilakukan validasi dan pengujian dengan menggunakan data yang sama, yaitu 99 data latih dan 25 data uji.

Pada *Backpropagation* validasi dan pengujian dilakukan dengan menggunakan parameter optimal yang telah diujikan sebelumnya, yaitu *Backpropagation* dengan jumlah *neuron* pada lapisan tersembunyi 5, besar *learning rate* 1 dan jumlah iterasi sebesar 5.000 iterasi. Sedangkan pada Algoritme Genetika – *Backpropagation* parameter yang digunakan adalah 90 jumlah generasi, 20 ukuran populasi, 0,1 probabilitas *crossover*, 0,9 probabilitas mutasi, 13 *neuron* pada lapisan tersembunyi, 1 nilai *learning rate*, dan 500 iterasi pelatihan *Backpropagation* dalam Algoritme Genetika. Tabel menjelaskan hasil validasi dan pengujian pada metode *Backpropagation* dan Algoritme Genetika – *Backpropagation*.

Tabel 6.12 Perbandingan hasil validasi dan pengujian

Metode yang Digunakan	Rata-Rata MAPE Validasi	Rata-Rata MAPE Pengujian
<i>Backpropagation</i>	1,449077	1,851893
Algoritme Genetika - <i>Backpropagation</i>	1,077603	1,741747

Dari hasil validasi di atas metode Algoritme Genetika – *Backpropagation* terbukti lebih baik dengan rata-rata MAPE hingga 1,077603 jika dibandingkan dengan validasi dari metode *Backpropagation* yaitu sebesar 1,449077. Begitu pula pada pengujian Algoritme Genetika – *Backpropagation* mampu memberikan hasil yang lebih baik yaitu 1,741747 sedangkan metode *Backpropagation* hanya mampu memberikan hasil 1,851893.



BAB 7 PENUTUP

7.1 Kesimpulan

Dari hasil telah didapatkan pada penelitian prediksi nilai tukar rupiah terhadap dolar Amerika dengan menggunakan metode Algoritma Genetika – *Backpropagation* dapat ditarik kesimpulan:

1. Implementasi Algoritme Genetika – *Backpropagation* berhasil dilakukan dengan jumlah data latih sebanyak 99 dan data uji sebanyak 25, dengan tahapan inisialisasi populasi, *crossover*, mutasi, pelatihan setiap individu menggunakan *Backpropagation*, evaluasi, dan seleksi. Hasil pengujian yang telah dilakukan didapatkan nilai MAPE terbaik sebesar 1,575318 dan rata-rata MAPE sebesar 1,741747. Algoritme juga mampu melakukan validasi dengan MAPE terbaik sebesar 1,0004917 dan rata-rata MAPE sebesar 1,077603.
2. Implementasi Algoritme Genetika – *Backpropagation* mampu memberikan hasil pengujian dengan nilai MAPE terbaik sebesar 1,575318 dan rata-rata MAPE sebesar 1,741747. Algoritme juga mampu melakukan validasi dengan MAPE terbaik sebesar 1,0004917 dan rata-rata MAPE sebesar 1,077603. Masing-masing pengujian dilakukan 10 kali percobaan yang dilakukan pada 99 data latih dan 25 data uji. Parameter terbaik dalam melakukan pelatihan dengan menggunakan Algoritme Genetika – *Backpropagation* adalah:
 - a. Jumlah generasi = 90
 - b. Ukuran populasi = 0,1
 - c. *Crossover rate* = 0,1
 - d. *Mutation rate* = 0,9
 - e. Jumlah *neuron* pada lapisan tersembunyi = 13
 - f. *Learning rate* = 1
 - g. Jumlah Iterasi = 500

7.2 Saran

Pada penelitian ini penulis merasa banyak kekurangan. Maka dari itu untuk pertimbangan penelitian selanjutnya penulis memberikan saran sebagai berikut:

1. Penelitian selanjutnya dapat melakukan pengujian terhadap pemilihan parameter masukan yang digunakan salah satunya dengan menggunakan fungsi autokorelasi (*autocorrelation fuction*).
2. Penelitian pada implementasi Algoritme Genetika – *Backpropagation* kedepannya dapat menggunakan metode seleksi lain seperti *binary tournament* atau *elistism*.

DAFTAR PUSTAKA

- Al Huda, F., Ridok, A. dan Dewi, C. 2013. Peramalan *Tlme Series* Saham Menggunakan *Backpropagation Neural Network* Berbasis Algoritma Genetika. S1. Universitas Brawijaya.
- Asriningtias, S. R., Dachlan, H. S., dan Yudaningtyas, E. 2015. Optimasi *Training Neural Network* Menggunakan *Hybrid Adaptive Mutation* PSO-BP, Jurnal EECCIS. 9(1), pp-79-84.
- Bergen, J. V., 2010. *6 Faktor s That Influence Exchange Rate* [Online] Tersedia di: <http://www.investopedia.com/articles/basics/04/050704.asp> [Diakses 26 Maret 2017]
- Dewi, C. dan Muslikh, M. 2013. Perbandingan Akurasi *Backpropagation* Neural Network dan ANFIS Untuk Memprediksi Cuaca. S1. Universitas Brawijaya.
- Fitriah, A., & Abadi, A. M. 2011. Aplikasi Model Neuro Fuzzy Untuk Prediksi Tingkat Inflasi Di Indonesia. PROSIDING, (Matematika dan Pendidikan Karakter dalam Pembelajaran).
- Gernowo, R., & Ernitawati, Y. 2013. Prediksi Variabilitas Nilai Tukar Rupiah Indonesia Terhadap Dollar Amerika Serikat Berbasis Sistem *Inference Fuzzy* Jaringan Syaraf *Adaptif* sebagai Analisis Tingkat Perekonomian. *Jurnal Sains dan Matematika*, 21(2), pp-39-41.
- Khan, A. U 2008. Genetic Algorithm Based *Backpropagation Neural Network* Performs better than *Backpropagation Neural Network* in Stock Rate Prediction. All Saints' College of Technology. Bhopal.
- Kusumadewi, S. dan Sri Hartati. 2006. NEURO FUZZY: Intergrasi Sistem Fuzzy & Jaringan Syaraf. Yogyakarta: Graha Ilmu.
- Mahmudy, W. F. 2008. Optimasi Multi Travelling Salesman Problem (M-TSP) Menggunakan Algoritme Genetika. Seminar Nasional Basic Science V, FMIPA, Universitas Brawijaya, Malang.
- Mahmudy, W. F. 2013. Dasar- Dasar Algoritme Evolusi. Malang: Program Teknologi Informasi dan Ilmu Komputer. Universitas Brawijaya.
- Mahmudy, W. F., Marian, R. M., & Luong, L. H. 2014. Hybrid genetic algorithms for part type selection and machine loading problems with alternative production plans in flexible manufacturing sistem'. ECTI Transactions on Computer and Information Technology (ECTI-CIT), 8(1), pp.80–93.
- Mahmudy, W., Marian, R., & Luong, L. 2013. Modeling and Optimization of Part Type Selection and Loading Problem in Flexible Manufacturing Sistem Using Real Coded Genetic Algorithms. World Academy of Science, Engineering and Technology International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering, p.7.

Mansur, M. 2009. Pengaruh Tingkat Bunga SBI dan Kurs Dolar AS Terhadap Indeks Harga Saham Gabungan Bursa Efek Jakarta Periode Tahun 2000-2002. *Working Paper Accounting and Finance*, pp.1-10.

Mawaddah, N. K., & Mahmudy, W. F. 2006. Optimasi Penjadwalan Ujian Menggunakan Algoritme Genetika. *Kursor*, 2(2), pp.1-8.

Muchlas, Z., 2015. Faktor-Faktor yang Mempengaruhi Nilai tukar Rupiah Terhadap dolar Amerika Pasca Krisis (2000-2010). *Jurnal JIBEKA*. STIE Asia Malang.

Pramesti, R. A. & Mahmudy, W. F. 2016. Optimasi Fuzzy Inference Sistem Mamdani Untuk Prediksi Nilai Tukar Rupiah Terhadap dolar Amerika Menggunakan Algoritme Genetika. S1. Universitas Brawijaya.

Pratama, T. I. B. 1999. Metode Peramalan Memakai Jaringan Syaraf Buatan Dengan Cara *Backpropagation*, *Jurnal Teknologi Industri*. 3(2), pp.109-116.

Shioda, K., Deng, S & Sakurai, A., 2011. Prediction of Foreign Exchange Market States with Support Vector Mechine. *International Conference on Mechine Learning and Applications*, 10, pp.327-332.

Siang, Jong Jek. 2005. Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan Matlab. Yogyakarta: Andi Offset.

Subiantara, R 2015, 'Prediksi Pergerakan Harga Harian Nilai Tukar Rupiah (IDR) Terhadap Dollar Amerika (USD) Menggunakan Metode Jaringan Syaraf Tiruan Backpropagation'. S1. Universitas Dian Nuswantoro Semarang.

Sumarto, K. C., Cholissodin, I., & Data, M. 2016. Peramalan Nilai Tukar Rupiah (IDR) Terhadap dolar Amerika (USD) Menggunakan Metode Support Vector Regression (SVR) Dengan Simulated Annealing – Genetic Algorithm (SA-GA). S1. Universitas Brawijaya.

Suprayogi, D. A., & Mahmudy, W. F. 2015. Penerapan Algoritme Genetika Traveling Salesman Problem with Time Window: Studi Kasus Rute Antar Jemput Laundry. *Jurnal Buana Informatika*, 6(2). Retrieved from <http://ojs.uajy.ac.id/index.php/jbi/article/viewFile/407/455>

Tahmasebi, P., Hezarkhani, A., Sahimi, M. 2012. Multiple-point geostatistical modeling based on the cross-correlation functions. 16(3), pp.779-797.

Wibowo, T., & Amir, H. 2005. Faktor-Faktor yang Mempengaruhi Nilai Tukar Rupiah. *Kajian Ekonomi Dan Keuangan*, 9(4).

Widodo, A. W., & Mahmudy, W. F. 2010. Penerapan Algoritme Genetika pada sistem rekomendasi wisata kuliner. *Genetika (GA)*, 3(4), p.5.